TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MUHAMMAD AMMAR AHMED

MAXIMUM AND LIMIT POWER TRACKING ALGORITHMS IN PV APPLICATIONS

Master of Science Thesis

Examiner: Assist. Prof. Tuomas Messo
Examiner and topic approved by the
Council of the Faculty of Computing and
Electrical Engineering on 31 May, 2017

ABSTRACT

MUHAMMAD AMMAR AHMED: Maximum and limit-power tracking algorithms in PV applications.
Tampere University of Technology
Master of Science Thesis, 58 pages, 4 Appendix pages
June 2017
Master's Degree Programme in Electrical Engineering
Major: Smart Grids
Examiner: Assist. Prof. Tuomas Messo

Keywords: renewable energy, solar power, solar cell, boost converter, MPPT, P&O algorithm, limit PV power technique.

During the last couple of decades, methods of electricity production have been changing significantly around the world. The transition is happening from the conventional sources of energy towards the renewables. Solar power is one of the significant source of renewable energy to electricity production. However, there are some limitations to convert this energy coming from sunlight into electricity, since, we cannot extract the whole energy of the sunlight that a solar panel receives. There are two major issues regarding the solar power production: one is the fluctuating grid power due to the large and rapid changes in PV power generation and the other is quite low conversion efficiency. Therefore, addressing these issues is the main focus area of the thesis.

The solar cell is highly dynamic in nature and its performance depends primarily on the light intensity E and the environmental temperature T. It is not possible to connect the power generated by a photovoltaic PV panel directly to load or utility grid, rather it needs to be controlled and modified in order to make it useable. For this purpose, there are various types of DC-DC converters designed depending upon the user's requirements. However, for the grid-connected PV modules DC-DC boost converters are used along with DC-AC inverters to stabilize the output power and to improve the solar power quality. In this thesis, MOSFET switched DC-DC 15V to 24V boost converter is used to take power from PV module and feed the stabilized power to the battery load. In addition, the Texas Instruments TM4C123G Tiva Launchpad microcontroller is used to regulate and control the converter.

The other major problem with the solar generation is significantly lower conversion efficiency from solar intensity to electricity. The solar efficiency is lowest compared to that of the conventional forms of electricity production. There has been lots of research work done for the efficiency improvement. There might be two ways of significant efficiency improvement in the whole solar power production process. One is to develop the internal structure of the solar cell with various chemical combinations of the material. The other method of efficiency improvement is the implementation of maximum power point tracking MPPT algorithms. In the V-I curve of the PV module, there is an operating point where the PV module produces maximum power with a specific temperature and sunlight conditions and this point of maximum power keeps on moving with changing climatic conditions. Therefore, this point needs to be tracked all the time and keep the operating

point of the PV system on it in order to get the maximum power out of PV module. In this thesis, the Perturb and Observe technique (P&O) is chosen as the MPPT algorithm.

To further enhance the performance of the PV system the power limiting technique, also known as the Constant Power Generation CPG, is used along with the MPPT algorithm. CPG basically limits the PV power generation on a certain value regardless of the total available PV power generation. The system calculates the generated PV power and compares it with the limit set by the user depending upon the value of the maximum power fed to the grid and the rated value of the components installed in the PV system. To ensure the maximum power generation within the limit-power value, system operates in MPPT mode, however, when the generated power reaches the limit, system automatically shifts towards the CPG mode. Likewise, when the available PV power values becomes less than the limit, system starts operating again in MPPT mode. Therefore, due to reason of the dynamic shifting of the operating point from MPPT to CPG and CPG to MPPT for tracking and generating PV power, the process is known as Hybrid Power Control (HPC). One of the major reason of implementing HPC is the safety and stability of the utility grid in case of the grid-connected PV system, since utility grids are effecting negatively due to increasing integrations of the renewables presently. However, limiting the PV power generation is also for the improvement of the utilization factor of the equipment installed that results in significant reduction of expenses. Another reason might be to avoid the chances of damage of the equipment, especially the switching circuitry, due to large peak to peak power variations and hence increase the overall lifetime of the PV system. The HPC feature is quite demanding nowadays especially in the grid-connected converter applications where solar power is being fed to the utility grid.

The Texas Instruments TM4C123G Tiva Launchpad microcontroller is used to regulate and control the converter operation. The hardware contribution of the Tiva Launchpad includes, generation of the PWM signal for the MOSFET gate terminal, and the measurement of the PV currents and voltages using ADC function. However, software work accomplishes the calculation of the PV power from measured PV current and voltage values and the implementation of the MPPT and the CPG algorithms. In this thesis work, the Code Composer Studio version 6.0 is the software used for coding and interfacing with the Tiva Launchpad.

PREFACE

This thesis has been written based on my research work at the department of Electrical Engineering in Tampere University of Technology from Nov 1, 2016 till May 25, 2017 under the supervision of Assistant Professor Tuomas Messo. The research work is related to the fields of power electronics and solar technology.

I would like to give special appreciate to my supervisor Tuomas Messo for his enthusiastic, encouraging and valuable support and continuous patience during the whole working period. All this done due to his efforts, guidance and quick feedback. In addition, I would say special thanks to doctoral student Aapo Apro for his precious time to help me a lot whenever I need during the thesis work although he was busy in his own PhD research work. He has provided with all the components needed in my practical work. I would like to pay my regards to Jyri Kivimäki to help me with the coding matters related to ADC conversion. In short, I would appreciate everyone from TUT who has helped me and encourage me directly or indirectly, to come up with this success.

Finally, I would say special thanks to my parents, family members and friends, especially my father, who has sacrificed all of his life to make me proud of this moment. Last but not the least; I would pray to Allah (God) for His countless blessings on me from the day one of my life up to this very moment.

Tampere, 31.5.2017

Muhammad Ammar Ahmed

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

AC	Analog Current
ADC	Analog to Digital Conversion
API	Application Programmable Interface
BIOS	Basic Input Output System
CCM	Constant Current Mode
CCS	Code Composer Studio
CPG	Constant Power Generation
CV	Constant Voltage
DC	Direct Current
DSP	Digital Signal Processor
EV	Electric Vehicle
FIFO	First In First Out
GPIO	General Purpose Input Output
HPC	Hybrid Power Control
I	Current
IC	Integrated Circuit
IDE	Integrated Development Environment
INC	Incremental Conductance
JTAG	Joint Test Action Group
LPC	Limit-Power Control
MPP	Maximum Power Point
MPPT	Maximum Power Point Tracking
OP AMP	Operational Amplifier
PCB	Printed Circuit Board
PE	Power Electronics
PLL	Phase Lock Loop
PV	Photo Voltaic
P-V	Power Voltage
PWM	Pulse Width Modulation
P&O	Perturb and Observe
RPC	Ramp-Power Control
SS	Sample Sequencer
uC	Micro-controller
V	Voltage

1. INTRODUCTION

This chapter of the thesis provides the general overview of all the electricity power generation methods used including conventional, non-conventional and renewable methods. Along with that, there is a detailed discussion about the solar technology, basic structure of solar cell, solar energy to electricity conversion efficiency and the challenges associated with the dynamic nature of a solar cell.

1.1. Methods of Electricity Production

Since the start of nineteenth century, there has been lots of various types of electricity power generation methods derived in order to fulfill the basic energy demands of human beings. As the time passed, the quality of life improved along with the increasing dependency on electricity and the growing electricity demand resulted in gradual increase in power production. Initially, only conventional methods of electricity generation was used including coal, natural gas and oil around the world and the contribution of coal to electricity generation remained highest compared to all other methods. However, nuclear energy and hydropower generation methods are derived later on [1]. A few decades back, the world started facing the environmental issue of global warming as a result of unwanted and intoxicated waste production and the extensive CO_2 gas emission in the form of smoke and other waste materials mainly coming out of the large industrial areas and power plants running with coal, natural gas and oil [2].

As the effects of global warming, the overall temperature of the world started increasing along with the unexpected and rapid climate change on our planet. Therefore, the focus of electricity production has shifted from the conventional methods towards the non-conventional methods of electricity generation. Besides global warming, another reason of this transition was the unavailability of natural reserves to meet the increasing demand of electricity. From the known figures of fossil fuel reserves given by International Energy Agency in 2009, the total coal reserves are around 909 billion tones and sustain for around 155 years in future with the same production speed, natural gas is estimated to have around 850,000 km^3 reserves and can remained for next 100 to 250 years with the current usage rate. Similarly, the oil reserves will run out after 100 years if we continue digging with the same rate [2]. Therefore, it would be quite critical issue after few decades in future if there will not be any alternative methods for electricity generation.

The most developing non-conventional or renewable methods known for electricity generation are solar power and wind power although, hydropower is also included in renewable or sustainable methods of electricity generation. Other than these, biomass, tidal

waves and geothermal are also some of the non-conventional methods for electricity production. In table 1.1 there is a comparison of shares of sources for electricity production between years 1973, 2001 and 2012.

Table 1.1. *Comparison of shares of sources of electricity production [1], [3]*

Year	Fossil (%)	Nuclear (%)	Hydro (%)	Other renewables (%)
1973	75.1	3.3	21.0	0.6
2001	64.5	17.1	16.6	1.8
2012	62.3	15.9	16.2	5.6

It is clearly evident from the table 1.1 that the electricity generation from renewable sources of energy is becoming more and more popular especially from the start of this century. According to [2], by the year of 2014, the total electricity generation by means of renewable energy was 23% of the total global power production. However, by looking in detail the statistics regarding the contribution of all the sources of electricity generation in EU countries from the time span of 1990 to 2015 stated in the report of *Eurostat Statistics Explained (2017)* and shown in the figure 1.1, it is quite clear that the solar power is the most trending renewable source of electricity after wind power among all the sources of electricity especially from last one decade [29].

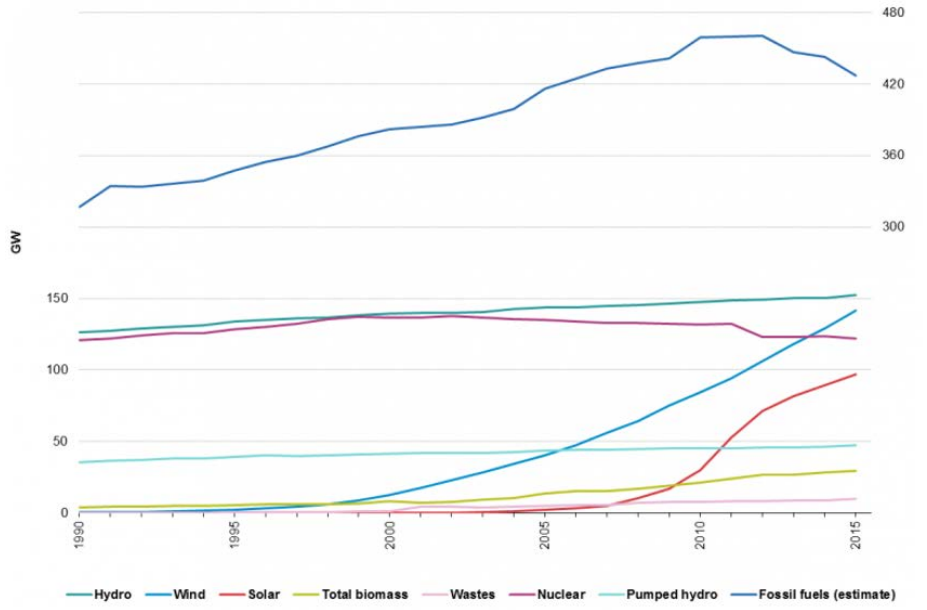


Figure 1.1. *Graphical Statistic of the electricity generation capacity in EU-28 from 1990 to 2015 [29]*

One of the reasons for the highest growth rate of the solar power is that it is possible to use at any scale from the large solar power parks having a rated capacity of megawatts connected with national grids to a small domestic level. Germany is one of the leading countries around the world having the highest domestic level solar power installations. [2] Another reason of might be the fact that sunlight is almost everywhere around the

world and solar power plant can be installed anywhere without any location and area limitations. [2]

1.2. Principle of Solar Power

According to a report presented by U.S department of Energy, the solar technology the world is utilizing for electricity production is not new. It might started from the 7th century B.C. when sunlight was utilized by magnifying glass to focus sunlight to fire and burn ants. However, in the last two centuries, French scientist A. Edmond Becquerel was the first person who discovered in 1839 that certain materials generate electricity when exposed to light. The latest shape we are experiencing presently has the origin from 1876 when W. G. Adams and R. Evans proved that when a solid material is exposed to light it start producing electricity without any moving body or thermal energy. The first experimental proof was given by R Millikan in 1916 related to the photoelectric effect. Later on, with further research, the first silicon photovoltaic cell was developed in United States with the light to electricity conversion rate of 4% and later in 1959, 11% efficiency was achieved by the same group of scientists. [4]

According to Dr. George (2016), depending on the usage and generation capacity there are three main types of solar power setups globally used;

- Off-Grid Solar Power Generation
- Grid-connected Solar Power Generation
- Hybrid Solar Power Generation

1.2.1. Off-Grid Solar Power Generation

Off-Grid solar generation is a type used mostly in low level generation setups e.g. at the roof of the houses. The power delivered by PV modules is first boosted up to the needed voltage value by DC-DC boost converter, then it is stored in battery banks and supplied to the DC loads and if needed, the stored power is inverted and fed to the AC loads. This kind of setups are used where electricity is needed to feed limited amount of load. Figure 1.2 shows the basic topology of Off-grid solar power systems.

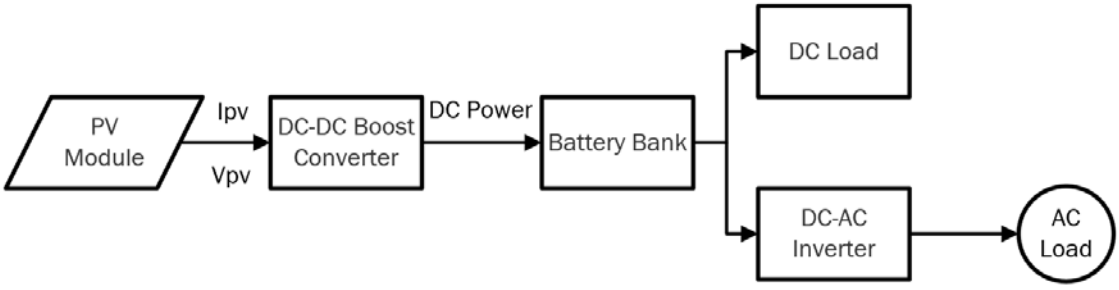


Figure 1.2. *Off-Grid solar power generation*

1.2.2. Grid-connected Solar Power Generation

Grid-connected solar power generation is mostly used with large electricity generation capacity at the national level production where solar power is transmitted to the utility grid. In this type of solar generation, utility grid is used as a load and besides DC-DC converter there needs to have a DC-AC inverter to transmit power to the grid. Most often, in large setups the DC solar power is inverted to three-phase AC power and then fed to the grid. However, one phase AC inverters are also used in the market. Figure 1.3 shows the brief picture of an on-grid solar power setup.

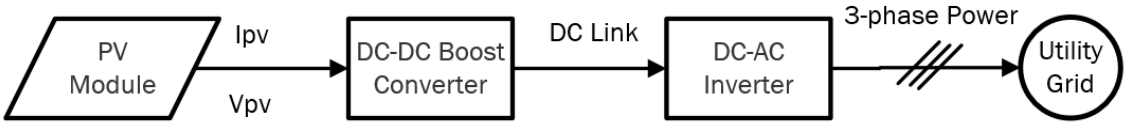


Figure 1.3. *Grid-connected three phase solar power generation*

1.2.3. Hybrid Solar Power Generation

Hybrid solar power generation is comparatively new concept of generation that is featured with both the On-Grid and Off-Grid generation qualities. The area of application for hybrid solar power is also quite large because it is useful for low capacity networks as well as medium capacity networks e.g. act as a micro-grid for large buildings, offices and shopping malls. In this type, the solar power form PV modules is first boosted up to the required voltage level and then divided into two paths; one is to charge the battery bank and the other path is to transmit to the utility grid. The whole process is automatically controlled by charge controller. Presently, we have quite advanced and self-intelligent charge controller that can sense the need of power on either side of the load i.e. DC load or utility grid, and can feed power to both sides simultaneously. However, we can also set a priority level in the charge controller depending upon our needs and requirements. In addition, we can also charge the electric vehicles EV through PV modules and if needed utilized EVs as DC battery bank to get the DC power. In figure 1.4 simple setup of hybrid solar power generation is described.

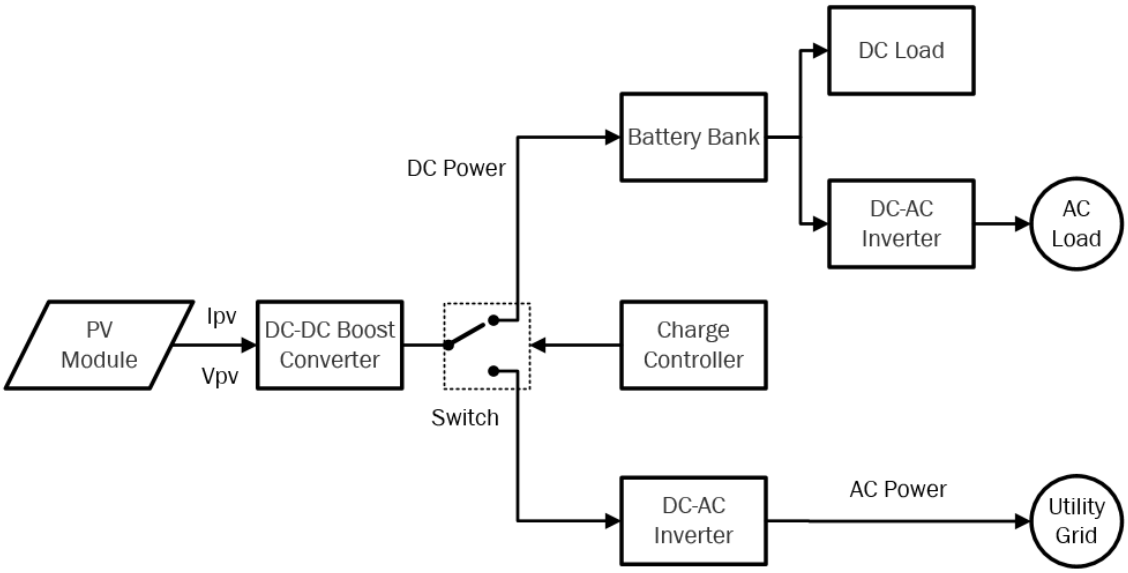


Figure 1.4. *Simple setup of hybrid solar power generation*

1.3. One-diode Model of Solar Cell

Silicon has the tendency to convert light photons, incident on its body, into electric charge. This behavior of silicon is known as the photovoltaic effect and in a solar cell there silicon is usually doped with phosphorus to make N-type materials or doped with boron to make it P-type material. However, in either case the impurity of other chemical element with silicon is one unit per million units of silicon [5]. During normal temperature (25°C) and irradiance (1kW/m2) conditions, the open-circuit voltage is located around 0.5 to 0.6 V for a crystalline Si cell and 0.6 to 0.9 V for an amorphous Si cells [13]. These solar cells are connected in series and parallel connections to make an array of PV modules to whatever voltage and current ratings are needed.

By observing the one-diode model, we can demonstrate solar cell as the current source connected in anti-parallel with a diode and shunt resistance. Figure 1.5 presents the one-diode model of a solar cell. In the circuit, current source represents the current i_{ph} drawn by the movement of electrons when sunlight is incident on the surface of the cell, however, the parallel connection of diode with the ideality factor A is the symbolic representation of the junction barrier between P and N type areas inside the material [5]. Therefore, when sunlight hits the surface free moving electrons break the junction barrier and enters into the P type region as represented by the diode current i_d. However, the series and parallel resistances r_s and r_{sh} presents the electrical losses inside the solar cell [2].

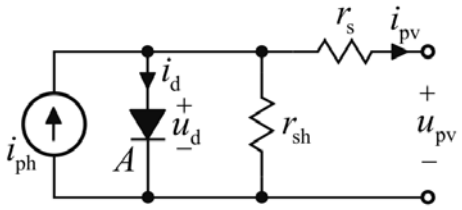


Figure 1.5. *One-diode model of a solar cell [2]*

The terminal voltage and the current is represented as u_{pv} and i_{pv} respectively. The current i_{pv} depends on to the amount of free moving electrons inside a solar cell which means that output current of a solar cell has a direct relation with the intensity of light incident on the surface. In addition, i_{pv} also depends upon the terminal voltage as depicted in IV characteristic equation below;

$$i_{pv} = i_{ph} - I_o \left\{ \exp\left(\frac{q(u_{pv}+r_s i_{pv})}{nKT_k} \right) - 1 \right\} - \frac{u_{pv}+r_s i_{pv}}{r_{sh}} \tag{1.1}$$

Where:

- I_o represents the reverse saturation current of the diode
- q represents the electronic charge
- T_k represents the temperature in kelvin scale
- K represents the Boltzmann constant
- n is a dimensionless factor

According to equation (1.1), the terminal current i_{pv} shows a non-linear behavior against the terminal voltage u_{pv} if we simulate an IV characteristic curve, according to [2], [6], as shown in figure 1.6

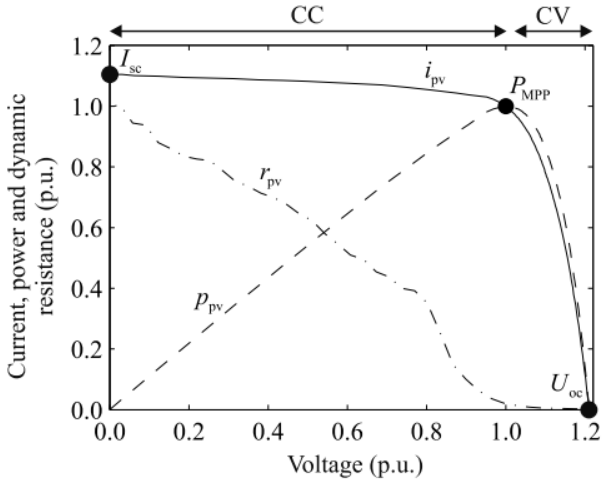


Figure 1.6. *IV-characteristic curve of a solar module [2]*

IV-curve depicts that the output current, known as short circuit current I_{sc}, goes to its maximum value when the voltage value is zero. However, the maximum voltage, known

as the open circuit voltage U_{oc}, appears across the terminals of PV module when the current goes to zero. There is a point in between these two peaks, as shown in IV-curve, where the solar module generates the maximum power according to the specific temperature and solar irradiance conditions, and the point is known as the maximum power point MPP on the IV-curve. This topic of MPP will be discussed later in more detail in the chapter 2.2. The IV-curve can be classified as two regions; the constant current CC and the constant voltage CV regions. In CC region, the current remains almost at same value before the MPP while voltage keeps on changing significantly, however in CV region, voltage values changes quite insignificantly after the MPP compared with that of current values as shown in IV-curve [2].

1.3.1. Dynamic Behavior with Light and Temperature

The non-linear terminal voltage and current behaviors, as shown in figure 1.6, clearly evident that the performance of a solar cell is mainly depends on the solar irradiance factor, climate temperature and also the load conditions [6]. The output current i_{pv} of a solar module depends on the solar irradiance as evident from equation 1.1 and shown in figure 1.7 (a). However, the terminal voltage u_{pv} mainly depends upon the environmental temperature as described in figure 1.7 (b).

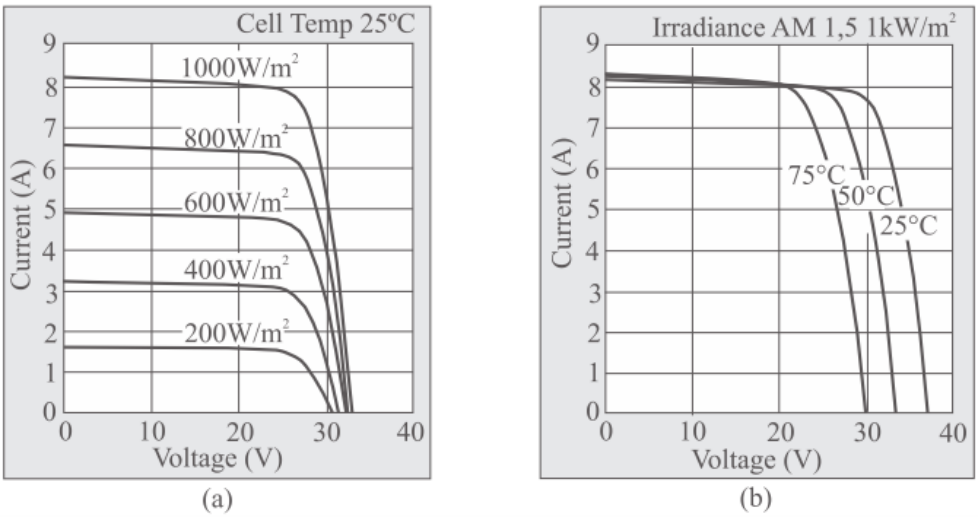


Figure 1.7. *IV-curve showing the dynamic behavior under a) constant temperature, b) constant irradiance and standard air mass index conditions [7]*

In real life as the temperature and irradiance conditions keep on changing all the time, therefore, the current and voltage parameters are also quite dynamic accordingly.

Moreover, there is another factor that causes changes in operating points especially the MPP of a solar cell is the amount of load connected with the solar modules [6]. For instance, when a heavy load is connected it tends to draw more current and to accommodate the effect output voltage value drops down as presented in equation 1.1 and vice versa happens when a light load value is connected.

1.3.2. Challenges Related to Climate Change

Although solar power is the most abundant and long lasting energy found on earth, there are certain challenges related to efficient use of this energy. As described in the topics above, changing temperature and irradiance are mainly the factors affecting the performance of a solar cell when exposed in real environment. Therefore, in order to make the solar cell perform well, we need to care about certain things and parameters.

The performance of solar power generation might depend on the regions of installation, e.g. in Nordic countries there is an issue of daylight sun as it reduces to almost zero during the winter times that affects the current values as presented in figure 1.7 (a). However, in African countries where the daylight and irradiance conditions are best suitable e.g., irradiance values goes up to 1400 to 1600 W/m^2, however, there are the challenges of high temperature which goes to around 55 to 60°C during the summer times and as a result the voltage values goes down according to figure 1.7 (b). [12] Therefore, it needs to be considered the suitable region where the temperature, irradiance and humidity…etc. conditions must be close to the standards.

In addition, there are certain challenges related to the internal performance of a solar cell that can be overcome by careful observation and controlling mechanisms. According to the PV curve presented in figure 1.6, there is a best suitable point for a solar module to operate i.e. MPP and it keeps on changing accordingly with the climate changes. We can mitigate this issue to let the PV module operate on that point all the time even it keeps on changing.

1.3.3. Efficiency of Producing Electricity

According to the report presented by PV Education Organization [8], the maximum solar cell conversion efficiency experimentally achieved is 41.1%, however, the experiment was done with the help of concentrator and the solar cell is the combination of $G_{0.35}In_{0.65}P$ materials. Although it was a great achievement for the improvement of solar efficiency, it is not practically implementable in real life. One reason might be the use of concentrator rather than natural sunlight that can help to improve the cell efficiency and secondly, $G_{0.35}In_{0.65}P$ material is quite expensive to be used at commercial level [8]. In spite of experimental test, most of the commonly used solar panels are made of mono crystalline and poly crystalline silicon materials. According to a research done by Gaur and Tiwari (2013) on different solar cells and calculated the efficiencies that are shown in table 1.2.

Table 1.2. *The conversion efficiencies of various material solar cells [9]*

Solar Cell Materials	Efficiency n_o(%)
Mono Crystalline Si	25.0
Poly Crystalline Si	20.4
CIGS[1]	20.3
Nano Crystalline Si	12.3
CdTe[2]	18.3

The solar modules made up of mono-Crystalline silicon material has the maximum efficiency of 25%. For instance if there are 100 units of solar energy available this PV module can convert only 25 units into electricity. It means, there is still lot of research work needs to be done to improve the conversion efficiency of the solar power generation.

1.4. Scope and Structure of the Thesis

The focus of the thesis is to research and develop the efficient designing of DC-DC boost converter and its hardware implementation. In addition, the implementation of Maximum Power Point Tracking (MPPT) algorithm using the Perturb and Observe (P&O) technique is done by using TM4C123G Tiva Launchpad micro controller DSP. However, the Limit power control concept also known as Hybrid Power Control (HPC) is also implemented to further improve the performance and the efficiency of the whole PV system.

The basic understanding related to the DC-DC boost converters, the MPPT and the limit power techniques will be explained in chapter 2. And the designing specifications and dimensions, and the DSP specifications will be discussed in chapter 3. Working procedures and principles, as well as simulations and results will be presented in chapter 4 and 5 respectively.

[1] CopperIndiumGalliumSelenide
[2] CadmiumTelluride

2. POWER ELECTRINICS FOR SOLAR POWER GENERATION

Power Electronics (PE) is involved in almost every kind of renewable energy resources including wind power and solar power, which are the most developing methods of electricity production [10]. PE has quite a vital role in the advancement of the solar technology. As illustrated in figure 2.1, starting from the PV modules, DC-DC step-up or step down converters, DC-DC links, DC-AC one phase or three phase inverters and also connection with grid and other loads, PE is involved in every segment of the whole process.

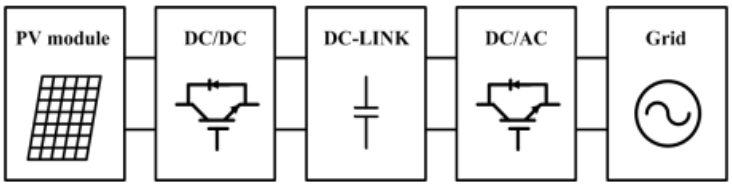


Figure 2.1. *Typical working process of a grid connected PV system*

2.1. Maximum Power Point Tracking

Maximum Power Point Tracking (MPPT) is an effective algorithm in PV applications to improve the solar energy to solar power conversion efficiency. In the IV curve, as shown in figure 1.5, MPP is where the product of PV output current i_{pv} and voltage u_{pv} is maximum. Since this point is non-linear and keeps on changing depending upon the climate conditions, therefore it needs to be tracked continuously and operate the PV module on this MPP. This whole process of tracking the point and operating the PV system on maximum power point is known as MPPT [7]. Figure 2.2 presented the P-V curve of a solar module where the generated power is plotted vertically and output voltage is plotted horizontally.

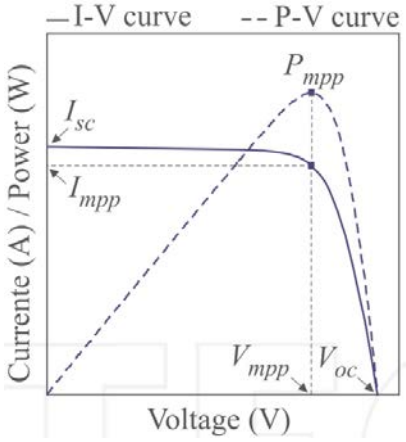


Figure 2.2. *P-V characteristic curve of solar module [7]*

There are two main categories involved in the tracking of MPP and operating the PV module on MPP. One could be related to the DC-DC converter type and type of load connected to the PV source and the other one could be the software, or the MPPT approaching techniques.

2.2. Importance of DC-DC Converter in PV Systems

Since the MPPT is keeping the operating point on the MPP, however, the operating point could be defined as the intersecting point of IV curve with the load curve of a system [7] as illustrated in figure 2.3.

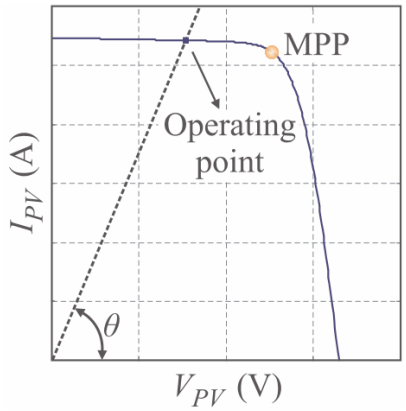


Figure 2.3. *Description of a system operating point*

Therefore, to keep the load curve on the MPP all the time, it needs to change the load values which is not possible because load is a defined value, however, MPP is quite dynamic in reality because once the climate conditions changes, the MPP also changes accordingly. This problem could be solved by using a suitable DC-DC converter, although DC-DC converters are used in PV application due to some other reasons as well [7]. By changing the duty cycle ratio, the load values are effectively changed in accordance with the PV voltage and current. In the Following sections, DC-DC converters are explained in detail in accordance with their applications in PV systems.

2.2.1. DC-DC Buck Converter

Buck converters are generally used in low power solar generation systems where output power needs to be stepped down and stored in the battery bank and then supplied to the connected loads [7]. Figure 2.4 shows the basic circuit layout of the MOSFET switched buck converter typically used in PV applications. A discrete Pulse Width Modulated PWM signal with a specific duty ratio (ranges 0 to 1) is given to the gate terminal of the MOSFET which acts as a switch in the circuit. The expression for the duty ratio is presented in 2.1 with the continuous current mode CCM operation of the buck converter and it describes the direct relation between input voltage and duty ratio, i.e. to decrease the output voltage value, the duty ratio value needs to be decreased. The duty ratio value is always between zero and one.

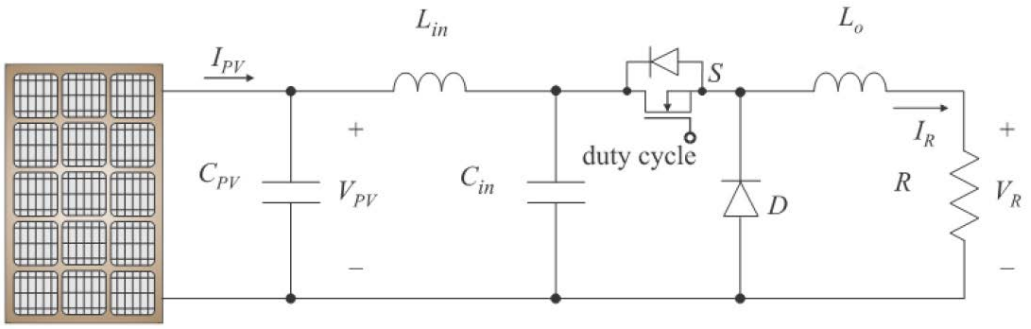


Figure 2.4. *Basic circuit of a buck converter [7]*

$$D = \frac{V_R}{V_{pv}} \tag{2.1}$$

2.2.2. DC-DC Boost Converter

Boost converters are generally used in grid connected PV applications where it needs to increase the output voltage up to 120V or 240V to transmit power to the utility grid. Figure 2.4 presents the on and off time circuits of a current fed DC-DC boost converter with MOSFET operating as a switch. For instance, if we try to draw the relation between the input voltage V_{pv} with the output voltage V_{out} for the whole switching period T_s, the waveforms appear are depicted in figure 2.5.

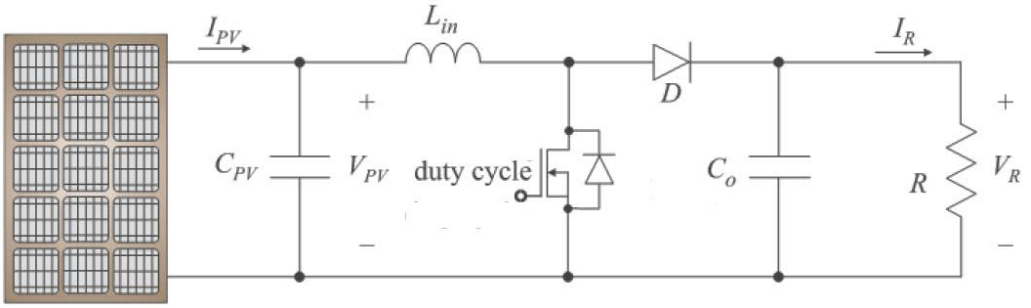


Figure 2.5. *Basic circuit of a boost converter [7]*

From the figure 2.6;

$$V_{pv}T_{on} = \left(V_{pv} - V_{out}\right) T_{off} \tag{2.2}$$

Or,

$$V_{out} = \frac{T_{on}+T_{off}}{T_{off}}\, V_{pv} \tag{2.3}$$

Since;

$$T_s = T_{on} + T_{off}$$

$$D = \frac{T_{on}}{T_s}$$

$D' = 1 - D$

Which implies;

$$V_{out} = \frac{1}{1-D} V_{pv} \quad \text{or} \quad D' = \frac{V_{pv}}{V_{out}} \tag{2.4}$$

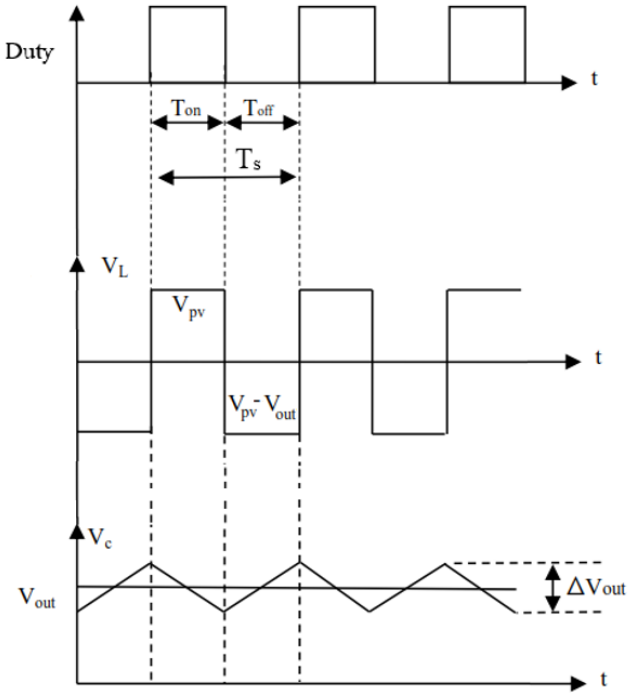


Figure 2.5. *Typical voltage waveforms of a boost converter [6]*

Therefore, equation (2.4) explains the inverse relation between the duty ratio and the PV voltage for a boost converter working in CCM. In this thesis work, boost converter is used and output voltage V_{out} is kept constant by connecting a DC battery load and hence only input voltage V_{pv} varies with varying duty ratio D of the MOSFET switch.

2.2.3. DC-DC Buck-Boost Converter

The buck-boost DC-DC converter topologies are typically used in Hybrid PV generation applications where sometime it needs to charge the batteries having low charging voltage values and sometime feed the utility grid. Therefore, buck-boost converters are applicable in such applications where it needs to feed a wide range of output voltages values. Figure 2.6 shows the electrical circuit diagram of the buck boost converters. Buck-boost converters operation is determined by the duty ratio value. The equation (2.2) describes that for the duty ratios less than 0.5, converter operates in buck mode and for the higher values from 0.5 to 1, converter operation is the boost mode operation. Therefore, we can say that the operational range of buck-boost is quite high with wide range of input voltage variations, compared with that of the buck and boost converters.

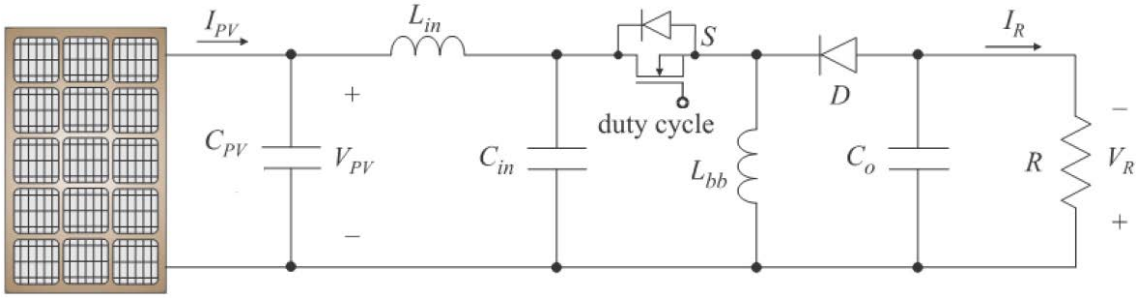


Figure 2.6. *Basic circuit of a buck-boost converter [7]*

$$D = \frac{V_R(1-D)}{V_{pv}} \tag{2.3}$$

2.3. MPPT Techniques

The other category for MPP tracking is the implementation of certain technique by using software-based implementation. There are various techniques that have been researched and developed to generate the maximum power out of PV modules, however, the most widely used techniques are; [14]

- Perturb and Observe (P&O)
- Incremental Conductance (INC)
- Constant Voltage (CV)

The P&O and INC are also known as the "Hill Climbing" techniques because they are implemented according to the differential ratios of power and voltage $\left(\frac{dP}{dV}\right)$. However, the CV technique is based on the ratio of $\frac{V_{MPP}}{V_{oc}} \approx 0.76$ [14]. Where V_{MPP} represents the voltage at MPP and the V_{OC} represents the open-circuit voltage. The P&O technique is explained in detail in the next section while the basic understanding of the other two techniques is illustrated below.

The incremental conductance IC technique calculates the deferential ratios of PV current and voltage $\left(\frac{dI}{dV}\right)$ to determine the direction of MPP on P-V curve, since at MPP, the $\frac{dP}{dV} = 0$ condition must be true. Therefore, an expression derived from these two ratios is illustrated in equation 2.4.

$$I + \frac{dI}{dV}V = 0 \tag{2.4}$$

The MPP tracker determines the MPP using the sign of the result calculated from the expression 2.4 [14].

The benefits of INC technique could be its rapid and accurate tracking of MPP with changing irradiance and temperature conditions. However, compared with the P&O technique INC is a bit complex especially in frequent climate changing conditions [14]. In addition, this complexity could be a reason to increase the computational time, and slows down the perturbing speed. [15]

The third most popular technique used for MPPT is the Constant Voltage CV. The CV technique is based on the fact that PV voltage does not depend most on the irradiance of a solar cell. Generally, the ratio of $\frac{V_{MPP}}{V_{oc}}$ is taken to be 76 percent, although it might be slightly different depending upon the solar cell properties of various PV modules [14], [15]. During the tracking time, the load is temporarily disconnected from the PV source and the PV voltage V_{OC} at open circuit conditions is measured. Then the operating voltage is set to the value 76 percent of the open Voltage value measured in open circuit conditions. [14] The same operating conditions remained for a specific amount of time and the tracker repeats the whole process again [14], [15].

The major flaw of this technique is the energy wastage during the time when the load is disconnected [14]. Moreover, it is possible that the MPP shifts from the operating point of 76 percent of theV_{OC}. [14], [15]. Another problem with this method could be the high fluctuations produced as a result of the frequent ON and OFF times of the connected load especially in large PV generators.

2.4. P&O Technique

The Perturb and Observe is the most common technique used in the MPPT algorithm due to its quite simple implementation, for instance, the whole setup needs to install two sensors, i.e. voltage and current sensor, in the circuit [16]. In this method, the duty ratio value of the DC-DC converter is adjusted to set the voltage V_{pv} value to the V_{MPP} and the current I_{pv} is changed accordingly [14]. In fact, the duty ratio of the converter changes the load impedance in order to match it with the impedance on the source side and maximum power is extracted (this concept is explained in detail in chapter 2.2). The relation for impedance matching is given in equation 2.5 [16]. The R_{eff} means the effective resistance

$$R_{eff} = \frac{V_{pv}}{I_{pv}} = \frac{(1-D^2)V_{load}}{I_{load}} = (1 - D^2)R_{load} \tag{2.5}$$

Since P&O is the hill climbing method, thus tracking is done by evaluating the present measured values of operating PV voltages V_{pv} and currents I_{pv} with the previous ones and the change in power and voltage is calculated. If the ratio of change in power over change in voltage is greater than zero, $\left(\frac{dP}{dV}\right) > 0$ then the tracker knows that it is moving in the right direction and continues its operation in the similar way. However, if the ratio gives

negative answer $\left(\frac{dP}{dV}\right) < 0$, it means that operating point is moving away from the MPP and tracker starts its operation in the opposite way. A detailed flow chat of the whole process in illustrated in figure 2.7

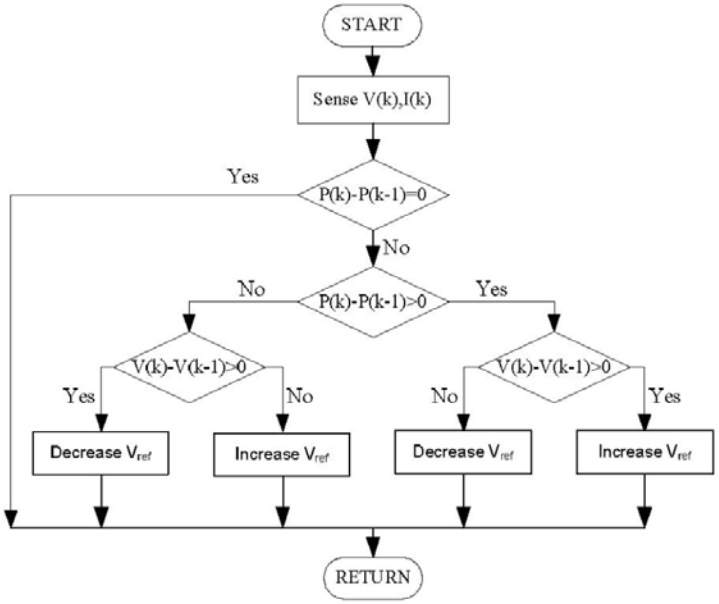


Figure 2.7. *The flowchart of the P&O technique for MPPT*

The complete cycle repeats in a specific interval of time thus, the tracker operation is periodic to converge the operating point to MPP on P-V curve [6], [16]. In figure 2.4 the term increase and decrease V_{ref} refers to the adjustment of the duty ratio in order to increase or decrease the V_{pv} value.

2.5. PV Power-limiting (Hybrid Power Control)

Since the integration of renewables, especially solar and wind powers, to the utility grid is increasing gradually as a result of the efforts made to reduce the issue of global warming [2], [17]. However, the problems related to these renewables are mainly the high fluctuations of voltages and power transferred to grid that results in bad quality of the whole grid power and the instability of the line frequencies [17]. Moreover, according to [17], following issues may arise as a result of increasing renewable sources of generation;

- Over-loading of the grid when the renewables power generation is at maximum.
- Low utilization factor of the PV inverters, instead of peak power generation time.
- Rapid and high fluctuations of power generation due to climate change may leads to degradation of the switching components and other devices.

There has been lots of solutions given in order to tackle the above issues. For instance, to mitigate the effects of overloading effect on the grid, [18] has presented the solution by installing battery banks to the output side of these generation plants, however, [19] suggested to expand the grid or integrate various grids to each other. However, both these

solutions are quite expensive with large initial installation cost. In addition, it requires extra monitoring and maintenance expenses [17]. Even though, if above suggestions are considered, the low utilization factor of the inverter and the rapid power fluctuations are still there. Therefore, it needs to have a better, reliable and affordable solution that considers all the above issues. In article [17], the solution has been presented that accomplishes all the problems and is known as the Hybrid Power Control (HPC) or the Limit-Power Control (LPC).

The main objective of HPC is to improve the thermal performance and the efficiency of the PV generation system and increase the utilization factor of the PV inverter [17]. Basically the HPC is the controlling and limiting of the PV power generated and transmitted to the grid.

2.5.1. Operating Principle

Hybrid Power Control (HPC) is the combination of two different controlling modes;

- Constant Power Generation (CPG)
- Maximum Power Point Tracking (MPPT)

There is a transition point between these two controlling modes, which is known as the power limiting point P_{limit}, that depends upon the over-loading conditions of the grid, thermal performance of the switching devices and the utilization factor of the PV inverter. Figure 2.8 illustrates graphically, the concept of P_{limit}. Since P_{pv} mainly depends on the solar irradiance and the temperature of the solar cell, therefore from time interval $t_1$to $t_2$and t_3 to t_4 the available power P_{pv} increased from the maximum allowable power to the grid or P_{limit}. As shown in the figure, $P_{limit}$is always less than the rated power of the PV system and it actually tries to limit the generation power at the point P_{limit}, although the PV is capable of generating more power P_{pv} as shown.

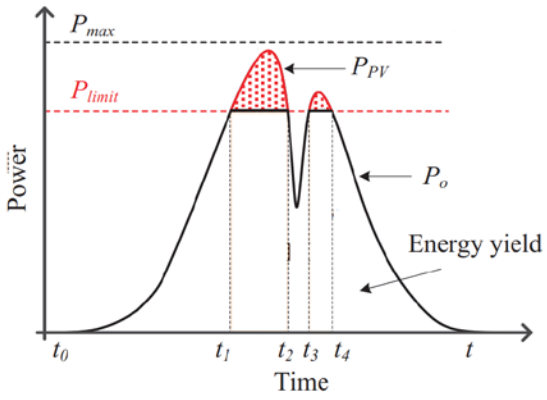


Figure 2.8. *Power curve of a PV system with HPC mechanism [17]*

Where P_o represents the actual power generation using MPPT algorithm and equation 2.6 is the mathematical expression for the P_o that explains mathematically the concept of HPC.

$$P_o = P_{pv}, \quad \text{when } P_{pv} < P_{limit} \quad \text{[MPPT controlling]} \tag{2.6}$$
$$P_o = P_{limit}, \quad \text{when } P_{pv} > P_{limit} \quad \text{[CPG controlling]}$$

According to [17], this P_{limit} must be an optimum point by taking into account various factors of PV thermal performance conditions, utilization factor of the PV converters and inverters, and the allowed annual loading conditions. Figure 2.9 shows the HPC controlling mechanism with the assistance of flowchart model.

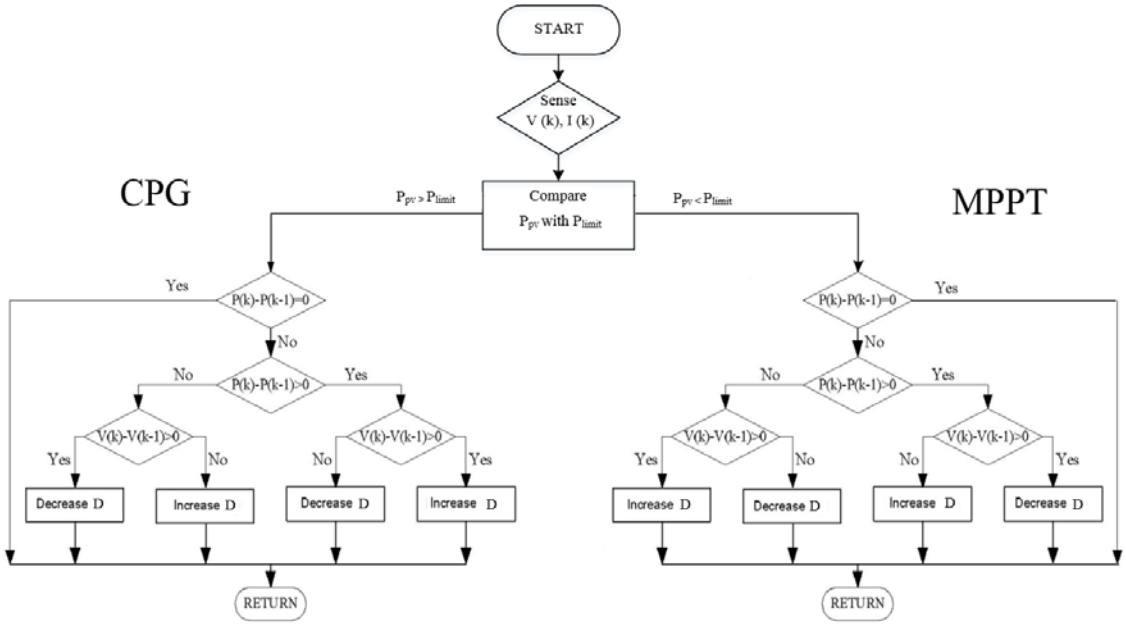
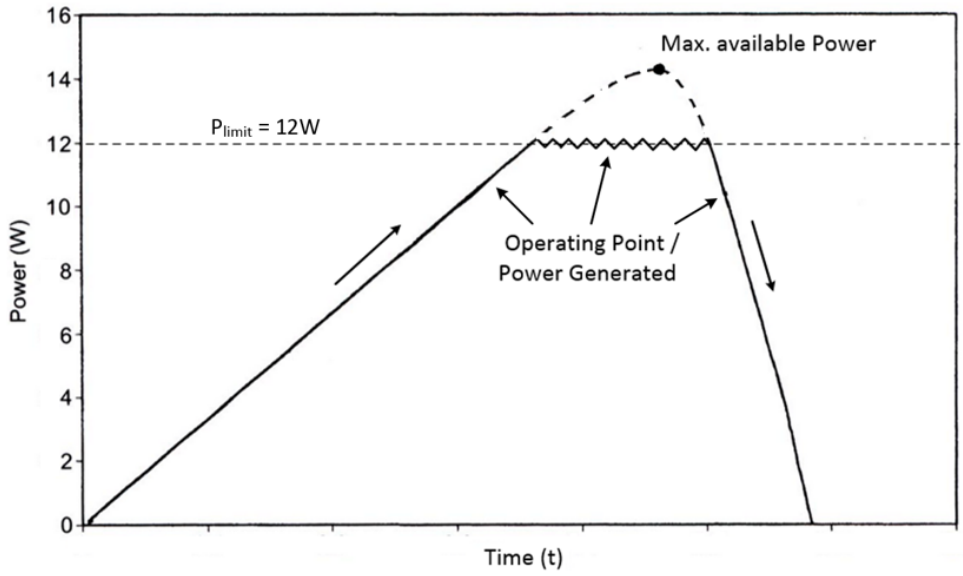
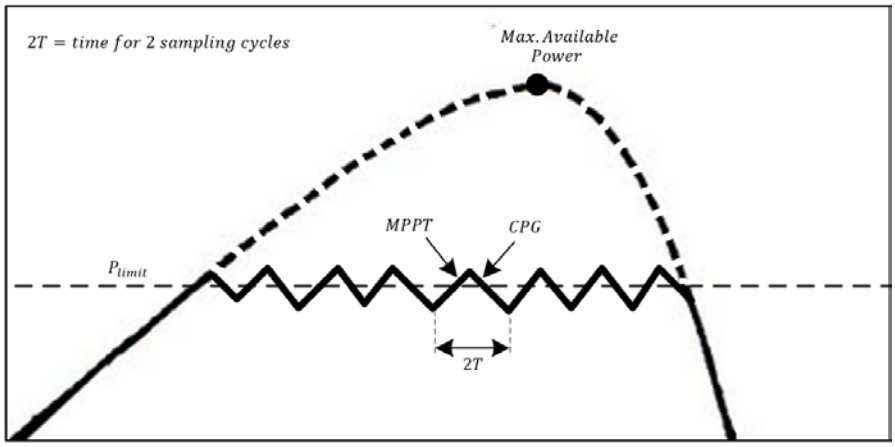


Figure 2.9. *Detailed flowchart of the hybrid power control mechanism*

Since, the HPC is the controlling the two different PV generation modes simultaneously as shown in figure 2.9 according to the limit power P_{limit} value. The process starts with the measurement of the actual power generation P_{pv} and then system compares that power with the P_{limit}value. If the generated power is greater than the set limit i.e. $P_{Pv} > P_{limit}$, then the system starts operating in the constant power generation CPG mode. In CPG mode, system tries to take the operating point away from the MPP and therefore, in CPG system works exactly in the opposite way as it works in the MPPT as illustrated in figure 2.9. Since the frequency of one complete sampling cycle is 2 Hz which means that after every 0.5 seconds of time system repeats the complete process of HPC. Therefore, once the CPG mode is selected then for the next 0.5 seconds the systems tries to operate away from the MPP points and as a consequence power generation P_{pv} value decreases. For the next sampling cycle, the P_{pv} value is more likely to be less than the power limit that leads to the system operate in MPPT mode and the process continues. Figure 2.10 explains the complete process illustrated in figure 2.9.



(a)



(b)

Figure 2.10. *PV Power generation curve with HPC mechanism*

2.5.2. Benefits of HPC

One of the major advantage of implementing CPG along with the MPPT algorithm is to improve the stability of the PV system as well as the utility grid [17]. Since there are large variations in the solar irradiance and the environment temperature in one complete day, typically the light intensity changes from 200 to 800 W/m^2 and the temperature varies from 20 to 40°C as shown in the figure 2.11 below. The experiment was taken during a typical American winter day in January 2013. However, even within the time period of hours there can be rapid fluctuations can be seen which implies that the power generation graph would also be more or less similar to the solar intensity graph in figure 2.11. These fluctuations might cause severe instability in the grid voltage and the line frequency. The HPC algorithm in PV systems tries to stabilize the PV power generation which consequently results in improved stability of the grid [25].

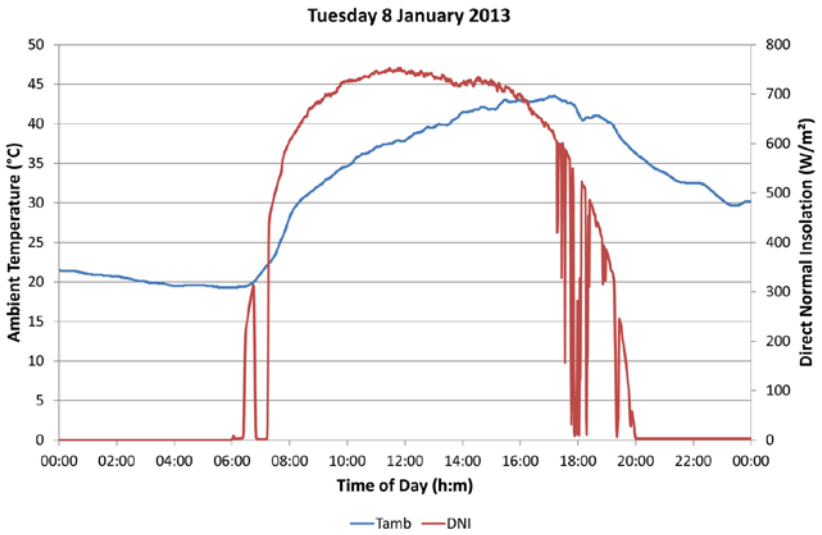


Figure 2.11. *Temperature and irradiance variations in a complete day [26]*

In addition, the utilization factor of all the equipment installed in PV generation system improves in HPC mode, especially the converters and inverters. Since the relation of utilization factor in hours is [17];

$$r = \frac{E}{P_{max}} \tag{2.7}$$

Where E is the annual energy production and the P_{max} is the maximum rated power of the PV plant. The relation in 2.7 tells the operating time in hours when the PV system operates at its maximum operating capacity over the period of one year. However, when the P_{max} value decreases to P_{limit}, the utilization factor of the system increases and as a result considerable amount of money could be saved by implementing the system equipment of less rated capacity [17].

Furthermore, the large and rapid changes in PV generation power, as a result of changing intensity as shown in figure 2.11, might results in increased wear and tear of the PV system equipment and consequently the life-time of the system reduces [17]. The HPC limits the power generation and reduces the peak to peak variation length of the PV power generation. Therefore the durability of the system increases [17].

2.6. Tiva Launchpad DSP

The importance of embedded systems is gradually increased with the advancement of power electronics especially in the renewable power generation networks. The most commonly used micro-controllers (uC) in embedded system applications are Arduino series, Raspberry pi series and the Tiva Launchpad series uCs. Raspberry pi uC are desined for the most advanced functinalities compared with the other two series. However, Arduino series is specially designed for the beginner level users and the interface is quite user friendly and simple. Tiva Launchpad series manufactured by the

Texas Instruments has large number of models with further sub-series of its uCs in order to accomplish a large number of various applications. Vast variety of uCs introduced above are used to efficiently drive and control the working of the PE equipment installed in PV and wind power genration systems.

The DSP board used in this thesis is the Texas Instruments Tiva lanchpad TM4C123G series uC. The purpose of emplying this uC is to regulate and control the power electronic circuitary involved and specially for the purpose of MPPT and HPC algorithms. The complete name of this model is;

Texas Instruments Ek-Tm4c123gxl Tm4c123g, Launchpad, Tiva C, Eval Kit

Figure 2.12. *The DSP board of the Tiva Launchpad TM4C123G uC [20]*

It is 40 functional pin IC uC with three phisical tiptop switches to reset and other functions, one power ON/OFF botton and two USB ports. One USB port is for the power and interfacing with PC to debug a program. A 16 MHz oscillating crystal in installed that drives a 400 MHz PLL. A default *divider /2* is already programmed to divide PLL, since it can only run at 400MHz [21]. To further decrease the frequency of PLL another *divider /5* is implemented to make a total dividing value of */10*. This makes the system to run at 40 MHz.In this thesis work, the uC is operated at 40 MHz of system clock cycle.

However, among a large number of functionalities of this uC, which can be found in its data sheet [20], following are the major functions utilized in this thesis work;

- Analog to Digital Conversion ADC
- Pulse Width Modulated PWM generation
- 3.3V and 5V pins as a source

2.6.1. Analog to Digital Conversion ADC

Since the function ADC converts the continuous analog input signal into a discrete times digital one. According to the datasheet of uC, there are two identical ADC modules **ADC0**

and **ADC1** that can share maximum of 12 input pins and each of the module can operate independently. Figure 2.13 illustrates the block diagram of the ADC modules.

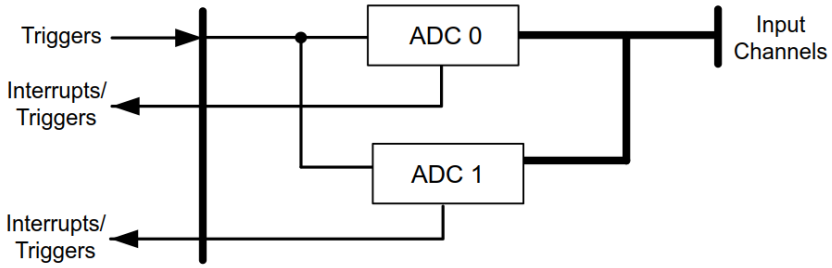


Figure 2.13. *Block model of the ADC modules [20]*

In addition, following features are added in each ADC function [21];

- 12-bit resolution
- On-chip internal temperature sensor
- Sampling speed of maximum one million samples per second.
- In each ADC, there are four sample sequencers each of them having different First In First Out (FIFO)s.
- Flexible Trigger control through, software, PWM, GPIO and Timers.

The function of sample sequencers (SS) is to collect and sample the input data. Each SS has various sampling depth or FIFO. The term sampling depth refers to the number of samples processed simultaneously. For instance, if an SS has sampling depth number of 4 then means that it has the capacity of sampling four data units at a time. Table 2.1 shows the sampling depth of each SS in an ADC module.

Table 2.1. *Details of the FIFO depth of each four sample sequencers [21]*

Sequencer	No. of Samples	Sampling Depth or FIFO
SS0	8	8
SS1	4	4
SS2	4	4
SS3	1	1

To configure an SS, the input source should be defined, request for triggering the specific sample and request for the interrupt generation after the sampling process has completed [20].

According to the datasheet, in the uC architecture the General Purpose Input Output (GPIO) pins are defined for all ADC 12 channels. Table 2.2 demonstrates these GPIO pins against each ADC channel. It is important to know these relationships, especially when configuring the GPIO pins for ADC in the coding phase.

Table 2.2. *List of GPIO Pins defined against ADC channels [20]*

Sr. No.	GPIO pin Name	ADC Channel
1	PE3	CH0
2	PE2	CH1
3	PE1	CH2
4	PE0	CH3
5	PD3	CH4
6	PD2	CH5
7	PD1	CH6
8	PD0	CH7
9	PE5	CH8
10	PE4	CH9
11	PB4	CH10
12	PB5	CH11

However, the uC reads the signal received at the ADC pins in the form of bits and since the TM4C123G has the 12-bit ADC resolution that means the bits range is from 0 to 2^{12} or 0 to 4096 bits [20]. In addition, according to the datasheet, the sensing voltage range of the ADC is from 0 to 3.3V. Therefore, the ADC resolution in the form of voltage can be calculated according to the equation 2.8;

$$ADC\ resolution = \left(\frac{1}{4096}\right) 3.3V = 0.8\ mV \tag{2.8}$$

The resultant solution of ADC resolution describes that voltage change of at least 0.8 mV can be sensible in the uC or with the change of 0.8 mV at the ADC pin, one bit changes in the uC ADC bits. In this thesis work, ADC pins PE3 and PE1 are used for the voltage and current measurements respectively and the block model shown in figure 2.14 describes the whole process of AD conversion.

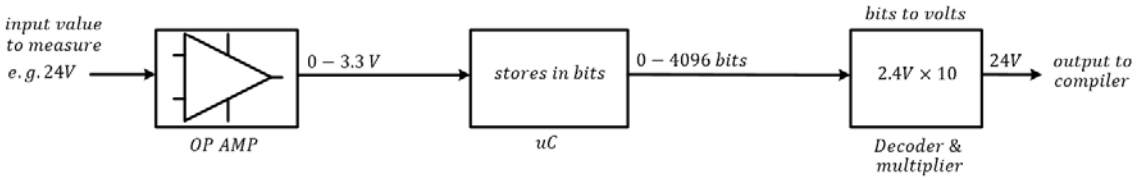


Figure 2.14. *Analog to digital conversion process in TM4C123G*

2.6.2. Pulse Width Modulation

Pulse Width modulation (PWM) is a periodically repeating digital signal used typically in switching power supplies and DC converters applications and controlling motor rotations [20].

In TM4C123G, two PWM modules are used, i.e. **PWM0** and **PWM1**. Each PWM module contains four PWM generator blocks and each generator further generates two independent PWM signals, which makes a total of 16 independent PWM signals that can be produced from this uC. A control block is also present that determines the polarity of the PWM signal and which signal is passed through to which pin [20]. The detailed block model is stated in figure 2.15 that explains the working of a PWM module.

Furthermore, each PWM generator has following features [22];

- Maximum PWM frequency rate of 40MHz can be generated.
- PWM signal can be down (trigger once in a complete cycle) or Up/Down (trigger twice in a complete cycle) count mode.
- One 16-bit counter
- Two PWM comparators
- Output signal of each PWM signal can be inverted.

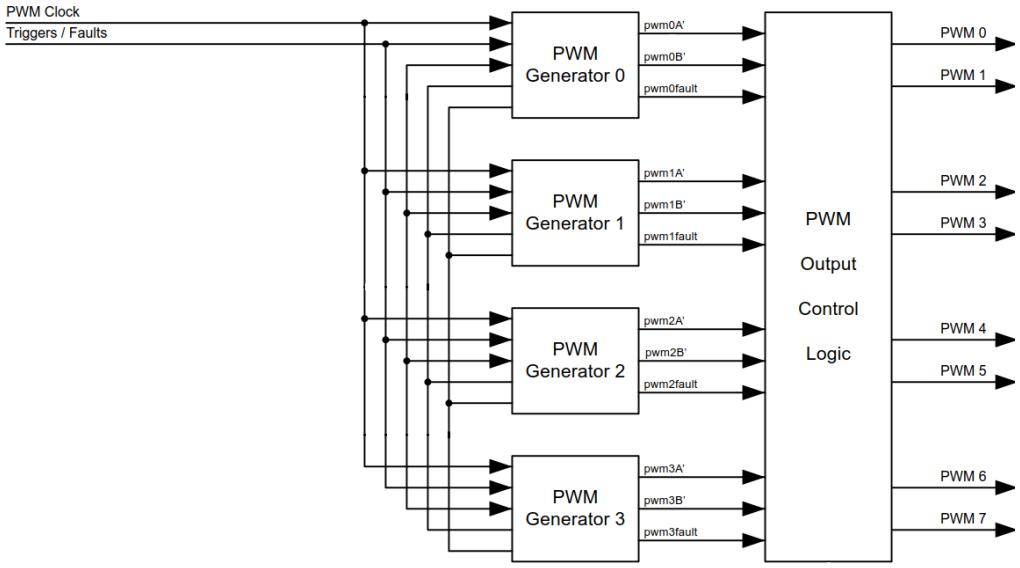


Figure 2.15. *Block model representation of the internal architecture of the PWM module [22]*

Since the system clock speed is 40 MHz, it is possible to run the PWM clock on different speeds by implementing various divider options Launchpad has provided. The available divider range is started from *divider by 2* and increases with the *divider by* 2^n up to *divider by 64* [22]. Moreover, in order to set the time period for one PWM complete cycle, it needs to set a counter value that the load register counts and triggers the PWM signal accordingly. Similarly to set the duty cycle, another value is required, also known as OCR value, in between the counter value maximum point and zero. Figure 2.16 illustrates the concept more clearly.

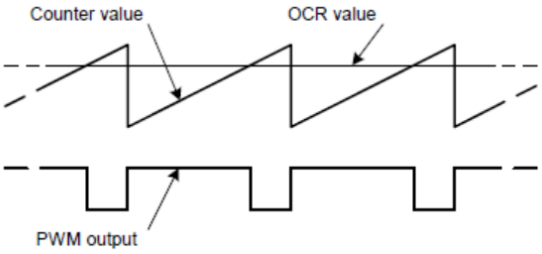


Figure 2.16. *PWM signal time period and duty cycle representation*

2.6.3. The Compiler Software

Code Composer Studio (CCS) is an Integrated Development Environment (IDE) compatible for the coding and debugging the program into uCs of Texas Instruments. Originally, CCS is designed for the debugging of embedded project designing and low-level Joint Test Action Group (JTAG) base applications. However, the latest versions are also including the OS level programming e.g. Android, Linux and Windows Embedded [23]. The syntax of the coding language used in CCS is quite similar with the C programming language. However, Code Composer Studio version 6.0 is used in this thesis work.

The Functional overview of how a program is established, compiled and debugged into the Launchpad in illustrated in as a flow chart in figure 2.17.

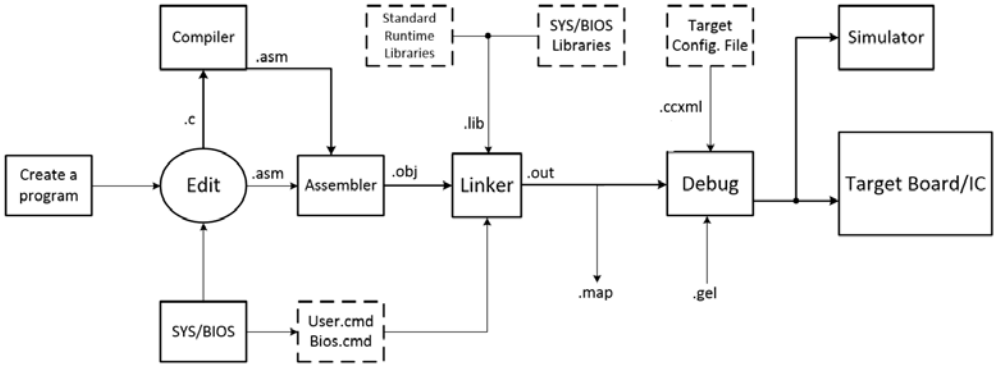


Figure 2.17. *Flow chart of the CCS functional overview [21]*

As depicted in the flowchart, the **.gel** file transfers instructions to the debugger related to the memory, and peripherals used in the program [21]. However, overall process includes compilers, assembler, linker, debugger, BIOS and the simulator.

The structure of a program in CCS is developed as follows;

1. Header Files or initializing the relevant libraries
2. Initialization of the global variables and external functions if any.
3. Main body of the program or main() function.
4. In main body, first set the speed of the system clock cycles and other clocks e.g. PWM clock.

5. Configuration of peripherals of the GPIO ports and other functions e.g. ADC or PWM.
6. Declaration of the specific pins e.g. PF3, to the specific functions e.g. M1PWM6
7. Enabling of the GPIO ports and the functions.
8. Couple of lines for the actual process of the functions used. E.g. setting the time period and duty cycle for PWM or setting the SS and input Channels for ADC.
9. Enabling of the modules utilized in the functions e.g. PWM generator 3 or the Sample Sequencer 2 for ADC.
10. Finally the infinite while loop that actually defines the actual process of the program after debugging into the target IC. Since the compiler remains here during the whole time execution, therefore the sequence of the instructions written in this loop is important.

For further understanding, it is recommended to go through the complete program code, written for this thesis work, in the Appendix B below.

3. DESIGN AND MODELING

Since the background of the thesis and discription of all the modules and major components used in practical work is already done in previous chapters, the actual design and modeling of the overall circuitary is discribed in this chapter. Figure 3.1 illustrate the circuit model representation of the DC-DC boost converter with PV module as a source, the battery bank as a load and the MOSFET as a switching device. The converter is regulated and controlled by the TM4C123G DSP by receiving PV voltage V_{pv} and currents I_{pv} values and generating the PWM signals for MOSFET.

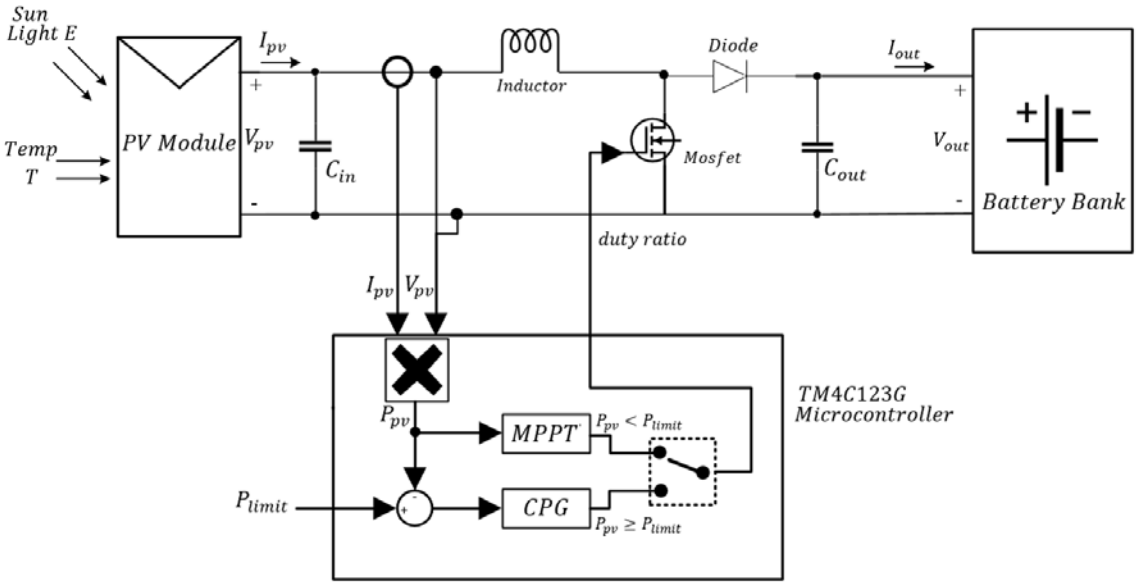


Figure 3.1. *Simplified circuit diagram of the overall working of the thesis*

Since Figure 3.1 shows that the uC defines which operation needs to be implemented depending upon the generation power P_{pv} and the limit power P_{limit} values. The detailed explanation of the working of boost converter and the uC will be discussed in chapter 4.

3.1. Parameters of the Designed Circuit

Table 3.1 and 3.2 explains the designing parameters and the specifications of boost converter and PV module used in the circuit shown in figure 3.1.

Table 3.1. *Parameters and specifications of Boost converter*

Output voltage V_{out}	24 V
Input voltage V_{pv}	8-17 V
Input current I_{pv}	0.95 A
Input Capacitor C_{in}	100 uF
Output Capacitor C_{out}	100 uF

Inductor	84.2 uH
Switching Frequency	150 kHz
Type of Switching Signal	Square wave
MOSFET	*2SK4017*
Diode	*SK56C*

Table 3.2. *Parameters of the PV module*

PV module Power Rating	
Open circuit voltage V_{oc} (at 900W/m^2)	20 V
Short circuit current I_{sc} (at 900W/m^2)	1.1 A
Expected MPP	76% of $V_{oc} \times I_{sc}$

3.1.1. Inductor Design

The designing of the inductor used in the boost converter circuit depends upon the type of the inductor core and its relative permeability, the copper wire resistance and the diameter, the switching frequency of the circuit and the value of the input capacitor used [24]. The inductance value L can be calculated using the expression (3.1) [24]. The peak to peak inductor current $\Delta i_{L,p-p}$ can be calculated as the 50% of the input current I_{pv}.

$$L = \frac{V_{pv} D T_s}{\Delta i_{L,p-p}} \tag{3.1}$$

Where;

- Duty ratio of the switching signal D
- Time period of the switching cycle T_s

The calculated inductance value is $L = 84.2\ \mu H$

In addition, the term permeability μ refers to the ability of a material to support the formation of a magnetic field inside its body. However, the relative permeability μ_e of a material is defined as the ratio of the permeability of air μ_o with that of the material. Equation (3.2) explains the relative permeability in symbolic form.

$$\mu_e = \frac{\mu}{\mu_o} \tag{3.2}$$

In this thesis work, the core material used for inductor design is N87 [24] and according to the datasheet, the relative permeability μ_e of the material is 2200. However, figure explains the magnetic circuit of an inductor with few other terms used in the inductor designing.

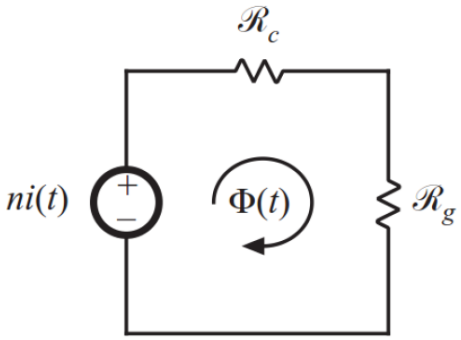


Figure 3.2. *The magnetic circuit of an inductor [24]*

Where;

- Material core reluctance: R_c
- Air reluctance R_g
- Magneto motive force ni

Since the permeability μ of the material is 2200 times larger than the air μ_o, therefore, the air reluctance R_g value is considerably larger than that of the material R_c [24]. However, equation (3.3) defines the formula to find out the number of turns of the inductor [24].

$$L = \frac{n^2}{R_g} = \frac{\mu_0 A_c n^2}{l_g} \tag{3.3}$$

Where;

- Inductance value L
- Permeability of air μ_0
- Area of the core A_c
- air gap length l_g

By putting all the parameter values, taken from the datasheet of the core material, the calculated number of turns for the indictor are: $n = 19.736 \approx 20$ turns.

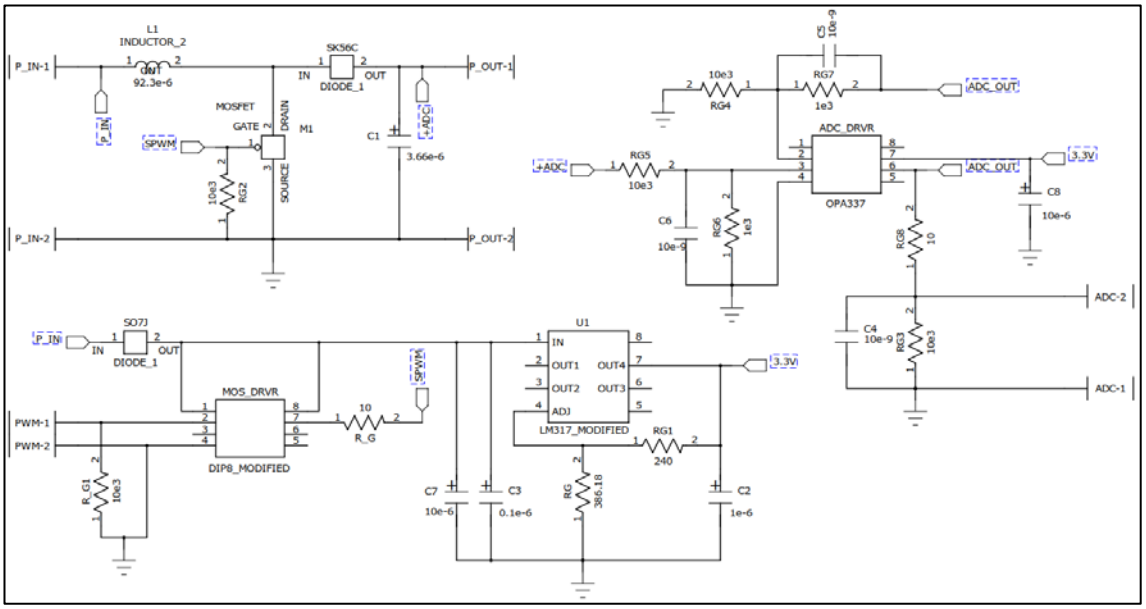
3.2. PCB Design

Physical designing and implementation is done on to the Printed Circuit Board (PCB). Therefore, the complete implementation procedure for the actual circuit could be divided into two parts;

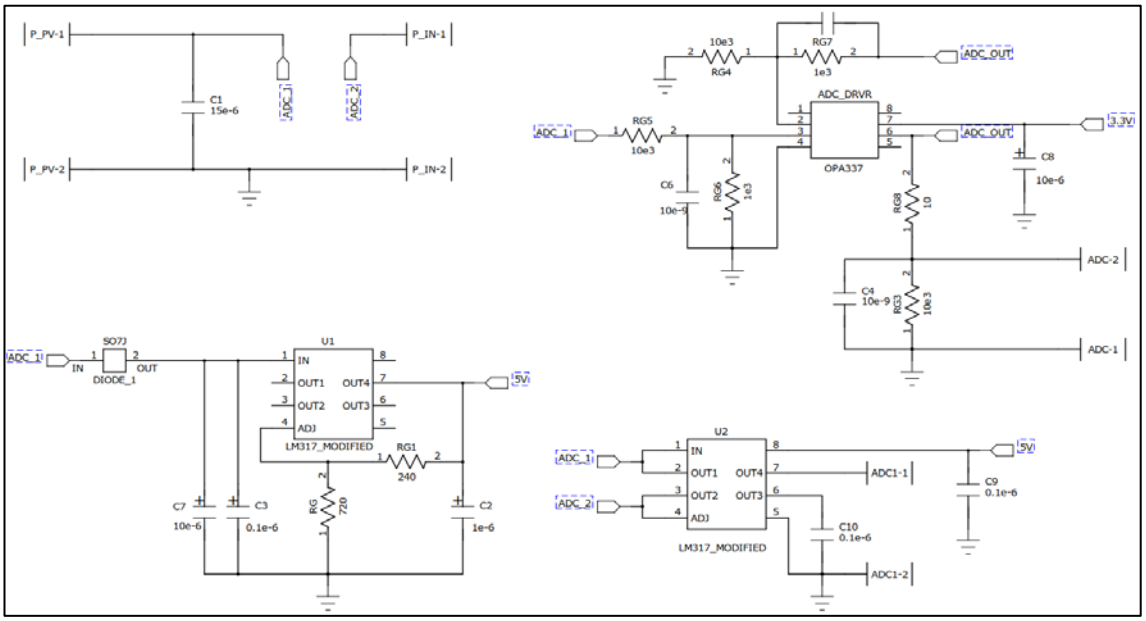
3.2.1. Software

The PADS Logic software is used for the designing the circuit with all the components needed. Figure 3.3(a) shows the designing of the boost converter in PADS logic contains an inductor, a Schottky diode, a MOSFET and an output capacitor. The components other than the main circuit includes a MOSFET driver MCP1407 to feed the PWM signal to the MOSFET gate terminal, an operational Amplifier (OP-AMP) OPA337 for measuring

the output voltage signal and a DC regulator LM317 in order to feed 3.3 volts to the OP-AMP circuit.



(a)



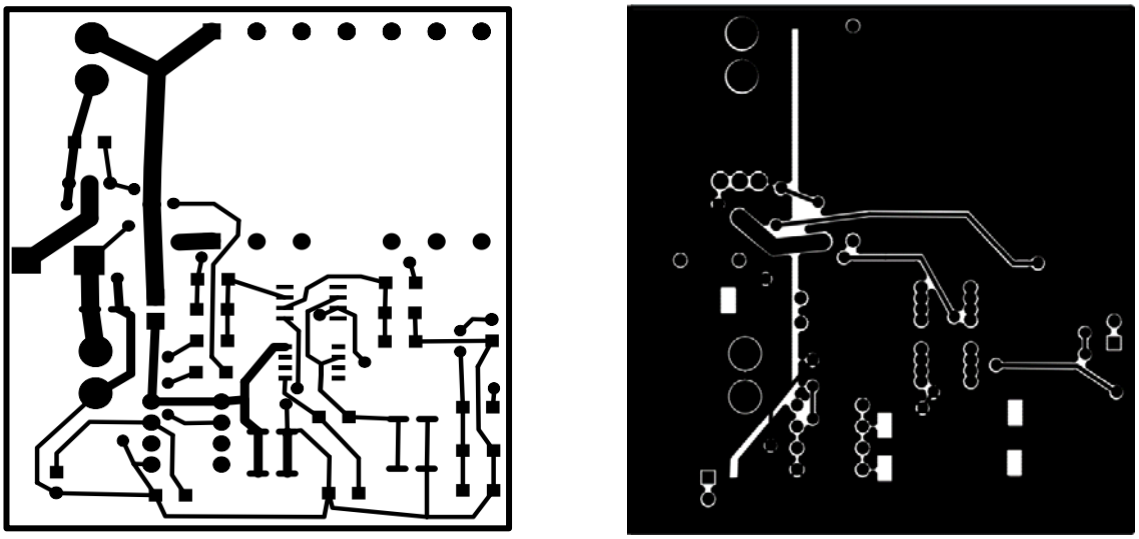
(b)

Figure 3.3. *The complete circuit diagram including all the components used. (a) represents the boost converter, (b) represents the measuring circuit.*

Since the input voltage of the boost converter needs to be controlled for the implementation of MPPT and CPG, therefore, figure 3.3(b) illustrates the measuring circuit implemented at the input side of the boost converter in order to measure the PV voltages V_{pv} and currents I_{pv}. The circuit includes an input capacitor, an OP-AMP OPA337 needs to measure the input voltage, a current sensor ACS712 used to measure input current in the

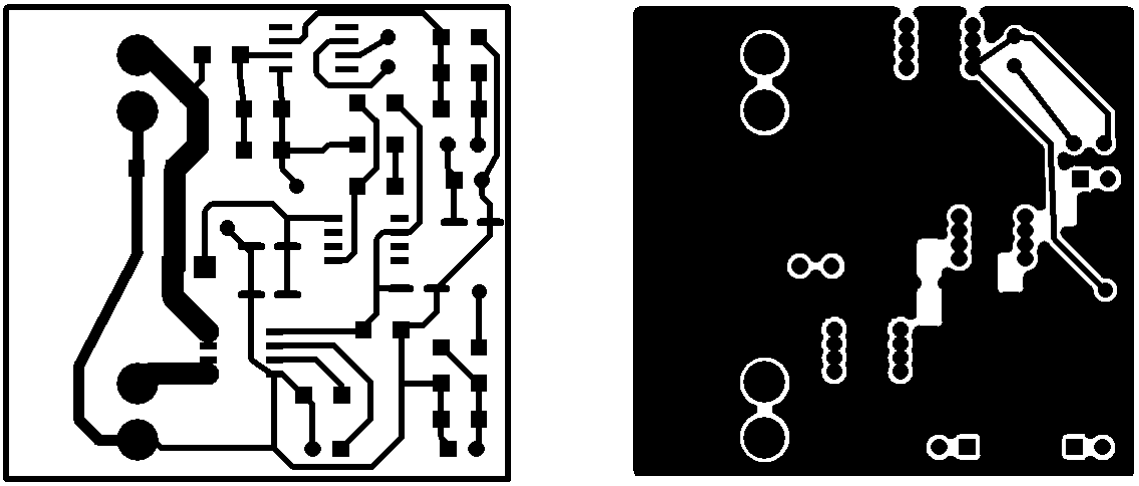
form of voltage using Hall Effect method and a DC voltage regulator LM317 to feed 5V to the current sensor.

After the circuit designing part in PADS Logic is completed, the whole circuit is sent to the PADS Layout where the PCB decals, of all the components used, needs to be defined and the final PCB Layout is designed. Figure 3.4 and 3.5 illustrate the PCB layout of the boost converter and the measuring circuit respectively.



(a) *(b)*

Figure 3.4. *The final PCB layout of the boost converter circuit with copper traces and a) shows the top side and b) side shows the bottom side of the PCB*



(a) *(b)*

Figure 3.5. *The final PCB layout of the measuring circuit with copper traces and a) shows the top side and b) side shows the bottom side of the PCB*

3.2.2. Hardware

In hardware design, the components are soldered to the copper printed PCB. Figure 3.6 shows the PCBs of both the boost converter and its measuring circuit with all the components attached.

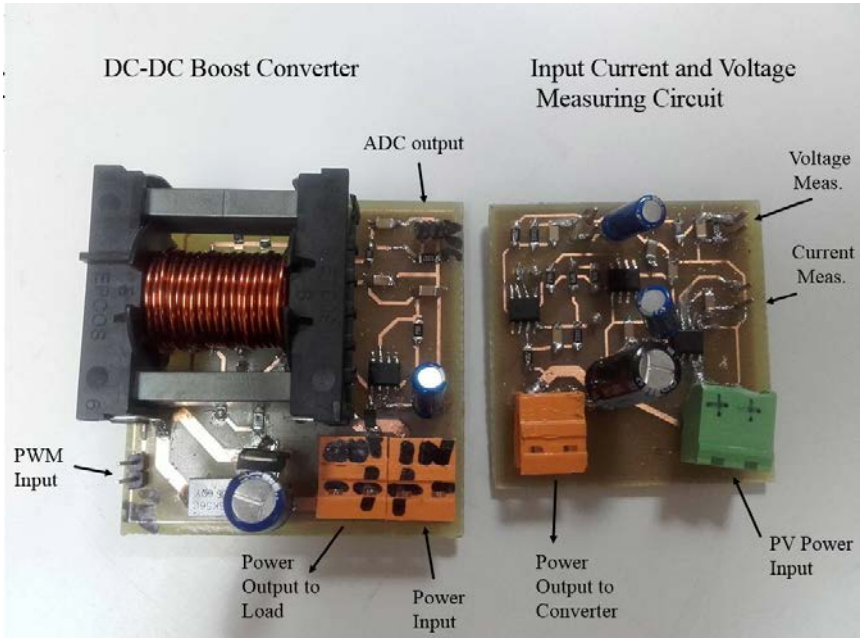


Figure 3.6. *PCB of complete circuit with all the components soldered*

On the left side of figure 3.6, the boost converter circuit board is shown with PWM input to MOSFET driver and ADC output from the OP AMP for the output voltage measurements. On the bottom side, power output terminal to battery load and the terminal of power input from measuring circuit is soldered. However, in the measuring circuit on the right side of the figure 3.6, output signals of measured input voltage V_{pv} and current I_{pv} are generated and on the bottom side, PV generated power input terminal and the terminal to the output power to converter are soldered.

4. WORKING PRINCIPLE

4.1. Boost Converter Open-Loop Design

For the boost converter open loop design, the steady state operating point expressions are derived and designed the boost converter simulation model in Simulink MATLAB. Figure 4.1 presents the general power stage of a voltage fed boost converter.

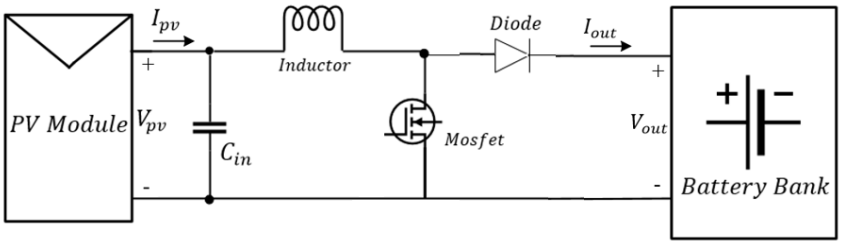


Figure 4.1. *Power stage of the voltage-fed Boost converter for PV applications*

ON time

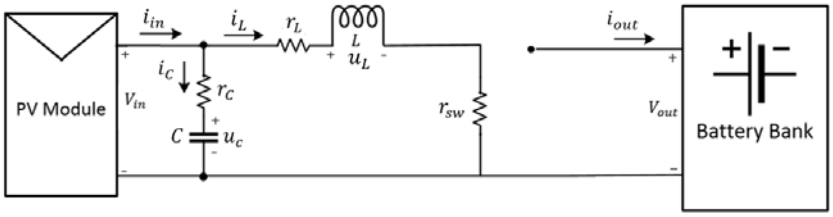


Figure 4.2. *ON-time sub-circuit of the boost converter*

Figure 4.2 presents the ON-time sub-circuit when the MOSFET is conducting with the drain-to-source resistance r_{sw} and diode is reverse biased. The ON-time inductor voltage, capacitor current, input voltage and output current equations are stated in (4.1)-(4.4).

$$v_{L-on} = v_c + r_c i_c - r_L i_L - r_{sw} i_L \tag{4.1}$$

$$i_{C-on} = i_{in} - i_L \tag{4.2}$$

$$v_{in-on} = v_c - r_c i_c \tag{4.3}$$

$$i_{o-on} = 0 \tag{4.4}$$

OFF Time

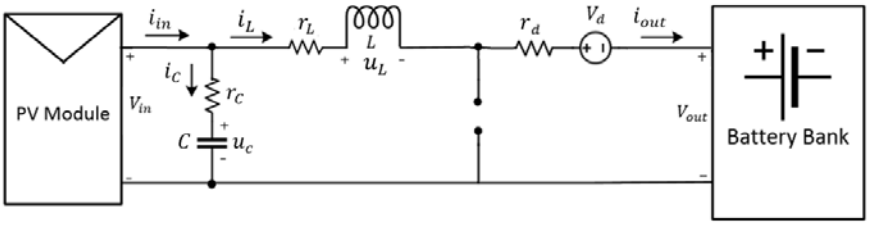


Figure 4.3. *OFF-time sub-circuit of the boost converter*

Figure 4.3 presents the OFF-time sub-circuit when the MOSFET is not conducting and diode is forward biased with forward voltage drop V_d and conducting resistance r_d. The ON-time inductor voltage, capacitor current, input voltage and output current equations are stated in (4.5)-(4.8).

$$v_{L-off} = v_C - r_L i_L - V_d - r_d i_L + r_C i_C - v_o \tag{4.5}$$

$$i_{C-off} = i_{in} - i_L \tag{4.6}$$

$$v_{in-off} = v_C - i_C r_C \tag{4.7}$$

$$i_{o-off} = i_L \tag{4.8}$$

Next step is to derive the averaged state-space model of the converter in CCM. The equations for the average state space values are written in (4.9)-(4.12)

$$\frac{d\langle i_L\rangle}{dt} = -\frac{r_c + r_L + dr_{sw} + d'r_d}{L}\langle i_L\rangle - \frac{d'}{L}\langle v_o\rangle + \frac{1}{L}\langle v_C\rangle + \frac{r_C}{L}\langle i_{in}\rangle - \frac{d'}{L}V_d \tag{4.9}$$

$$\frac{d\langle v_C\rangle}{dt} = -\frac{1}{C}\langle i_L\rangle + \frac{1}{C}\langle i_{in}\rangle \tag{4.10}$$

$$\langle v_{in}\rangle = \langle v_C\rangle + r_C\langle i_{in}\rangle - r_C\langle i_L\rangle \tag{4.11}$$

$$\langle i_o\rangle = d'\langle i_L\rangle \tag{4.12}$$

The steady state operating point needs to be calculated from the average state space model equations (4.9)-(4.12) by putting the left hand side derivative variables equal to zero and change the average valued variables with the upper case later. The derived expression for the steady state operating point in symbolic form is stated in (4.14)

$$0 = -(r_L + Dr_{sw} + D'r_d + r_C)I_L + r_C I_{in} + V_C - D'V_o - D'V_d$$

$$I_o = I_L D' \qquad \rightarrow \qquad I_L = \frac{I_o}{D'}$$

$$I_{in} = I_L$$

$$V_{in} = V_C - r_C I_L + r_C I_{in} \qquad \rightarrow \qquad V_{in} = V_C$$

That can be simplified as;

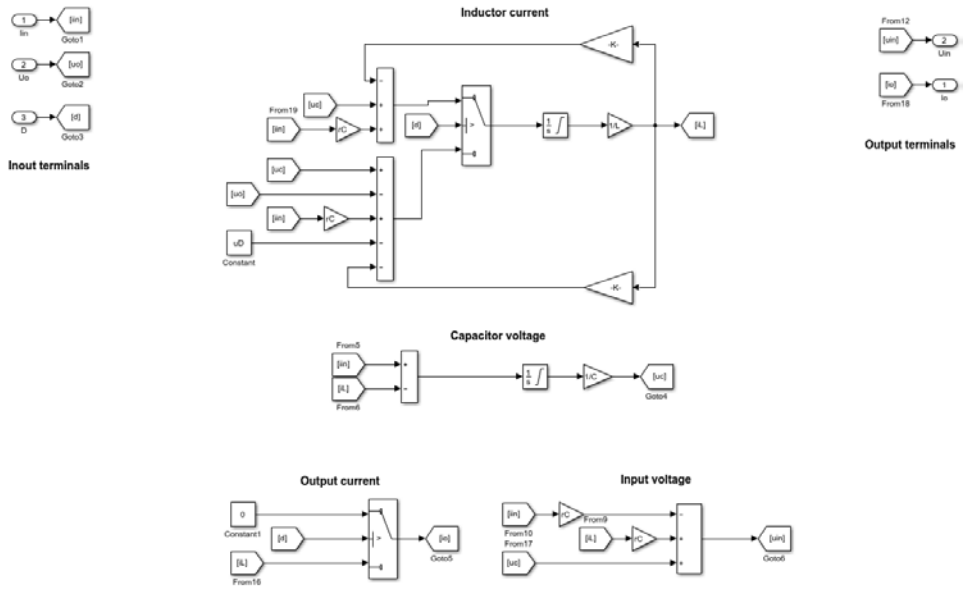
$$V_{in} + r_C(I_{in} - I_L) - r_L I_L - D r_{sw} I_L - D' r_{sw} I_L - D' r_d I_L - D' V_o - D' V_d \ = 0 \qquad (4.13)$$

Or,

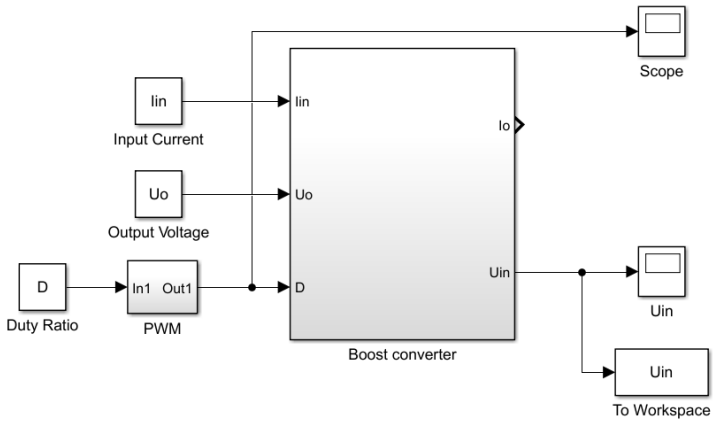
$$D' = \frac{V_{in} - (r_L + r_{sw}) I_{in}}{V_o + V_d - (r_{sw} - r_d) I_{in}} \qquad (4.14)$$

4.1.1. MATLAB Simulations

From the ON and OFF-time expressions derived in (4.1)-(4.8), the simulation model of the boost converter is designed in SIMULINK as depicted in figure 4.4.



(a)



(b)

Figure 4.4. *a) Subsystem of the boost converter, b) Simulink Model of the boost converter with open-loop testing parameters*

Since the boost converter is designed for open-loop test, therefore all the input parameters are either given or calculated using MATLAB. The duty ratio value D can be solved from the derived expression (4.14).

$$D' = \frac{V_{in}-(r_L+r_{sw})I_{in}}{V_o+V_d-(r_{sw}-r_d)I_{in}} = 0.6464$$

$$D = 1 - D' = 0.3536$$

Whereas;

$v_{in} = 16V$ Since the PV module used in this thesis is 20W rated power.

$I_{in} = 1.25A$

$r_L = 100m\Omega$

$r_{sw} = 70m\Omega$

$V_o = 24V$ Because the battery load voltage is 24V

$V_d = 0.5V$ Schottky diode forward voltage drop

$r_d = 10m\Omega$ Schottky diode forward biased resistance

The code for the boost converter open-loop test written in MATLAB mfile is;

```
%Open-loop control of boost converter
%PV panel spec: 20W, Voc = 21.5V, Isc = 1.25A
clear all
clc

%------Converter Design Parameters------%

fs = 150e3;      % MOSFET Switching frequency
Ts = 1/fs;       % Time period for one switching cycle
Uin = 16;        % Input voltage
Uo = 24;         % Output voltage
Iin = 1.25;      % Input current
L = 84.19e-6;    % Inductance value calculated in section 3.1.1
rL = 100e-3;     % inductor resistance given in the data sheet
rC = 10e-3;      % capacitor resistance given in the data sheet
rD = 10e-3;      % forward biased schottky diode resistance
uD = 0.5;        % schottky diode forward voltage drop
rsw = 70e-3;     % MOSFET drain-to-source conduction resistance
C = 100e-6;      % Capacitance value
s = tf('s');     % defining the laplace transfer function 's'

%-------Steady-state Duty Ratio-------%

D1 = (Uin - (rL + rsw)*Iin)/...    % duty ratio expression derived from
     (Uo + uD -(rsw -rD)*Iin);     % equation (4.14)
D =  1-D1;
```

The desired resultant input voltage waveform of the boost converter for the open loop in Simulink is shown in figure 4.5.

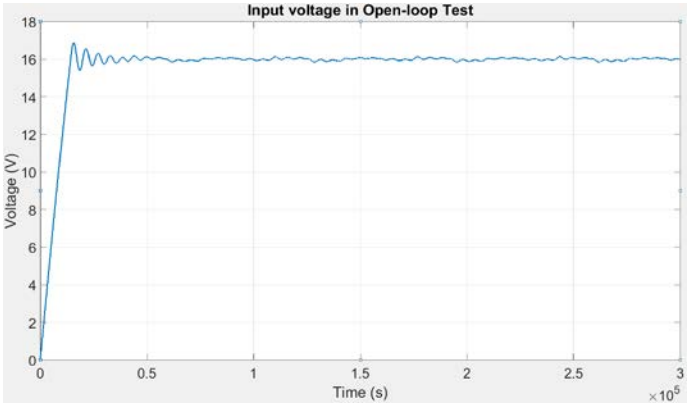


***Figure 4.5**. The input voltage waveform of boost converter in open-loop*

4.1.2. Code Composer Studio CCS

Since the DSP used in this thesis is Tiva Launchpad TM4C123G and the CCS is the compatible compiler software provides the work space for coding and burns the codes into the DSP. TivaWare™ is written using the ISO/IEC 9899:1999 (or C99) C programming standards along with the Hungarian standard for naming variables [21]. However, for the open loop test of the boost converter, it needs to generate a constant PWM signal from the DSP board to derive the operation of the MOSFET switch for the voltage boosting. However, for better understanding each line written in the CCS code for the open loop test is explained below;

Header Files

```
#include <stdint.h>
#include <stdbool.h>
#include "driverlib/gpio.h"
#include "driverlib/pin_map.h"
#include "driverlib/pwm.h"
#include "utils/uartstdio.c"
```

These are the initial lines of the code in CCS needed to access the TivaWare Application Programmable Interface (API) [20].

`stdint.h` and `stdbool.h` are the variable definitions and Boolean definitions respectively for the C99 standard [20]. In addition, `utils/uartstdio.c` is the "utils" folder in TivaWare APIs that contains functions for the UART "stdio" and `pin_map.h` is for the mapping of the used peripherals with the DSP pins. However, `driverlib/gpio.h`, and `driverlib/pwm.h` are to request the access of GPIO pins and the PWM function respectively from the `driverLib` library.

Defining Global Variables

Global variables refers to the variables that can be accessed in main function as well as the external functions written in a code.

```
float period = 133;
float duty = 67;
```

These variables are for setting the PWM time period and the duty ratio. Since the PWM clock is running at the 20MHz frequency and `period = 133` means that the one PWM period is equal to the time taken by PWM clock to complete 133 cycles and duty ratio is exactly half time that make the duty 50 percent of the period, since the code is written for the open loop test.

Main Function

```
int main (void)
{
}
```

These lines represents the main function body. The complier always remains inside the main function and in case of running any other function outside main function, it needs to call that function inside the main function.

Setting the Clock

```
SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYS-
CTL_OSC_MAIN|SYSCTL_XTAL_16MHZ);
```

The above line of code is to configure the system clock. Since, it is already explained in detail in the section 2.6 above, system clock runs at 40MHz by dividing the 200MHz PLL oscillations by 5.

```
SysCtlPWMClockSet(SYSCTL_PWMDIV_2);
```

To configure the PWM clock to 20MHz with reference to the system clock, it needs to divide the system clock by 2 as stated in the code line above.

Peripherals Enabling

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM1);
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
```

Before using any peripheral in the code, it needs to enable them in the `driverLib` function. Since the PWM function and the GPIO port F is used in this code, therefore the above two lines of code are for the enabling these peripherals. `PWM1` refers to the PWM module 1.

PWM Configuration

```
GPIOPinConfigure(GPIO_PF2_M1PWM6);
GPIOPinTypePWM(GPIO_PORTF_BASE, GPIO_PIN_2);
```

In the first line of above two lines, PWM module 1 PWM6 is selected, however in the second line, the GPIO pin PF2 is integrated with the PWM function.

```
PWMGenConfigure(PWM1_BASE, PWM_GEN_3, PWM_GEN_MODE_DOWN | PWM_GEN_MODE_NO_SYNC);
```

In the above line of code, PWM generator 3 is selected, in addition, the `PWM_GEN_MODE_DOWN` instruction selects the shape of the PWM signal to the saw tooth shape.

```
PWMGenPeriodSet(PWM1_BASE, PWM_GEN_3, period);
PWMPulseWidthSet(PWM1_BASE, PWM_OUT_6,duty);
```

These two lines configures the time period and the duty ratio variables defined above the PWM peripheral.

```
PWMGenEnable(PWM1_BASE, PWM_GEN_3);
PWMOutputState(PWM1_BASE, PWM_OUT_6_BIT, true);
```

The first line enables the PWM mode 1 generator 3, however, second line turns ON the PWM output pin.

Infinite While() loop

```
While (1)

{
PWMPulseWidthSet(PWM1_BASE, PWM_OUT_6, duty);
}
```

Finally in the main function, an infinite while loop is written in order to generate PWM signal for the infinite time. This loop is the actual working area of the compiler because any function used in the code needs to be defined here. Since, only the PWM function is used for the open loop test, therefore it is written in the while loop.

To justify the above written CSS code, there is a video demonstration of the open loop test of the boost converter present online on YouTube. The online link to the video is written below;

https://www.youtube.com/watch?v=xxQR78UHIzY

As illustrated in the video, the applied input DC voltage to the boost converter is 12V and the output voltage is boosted to around 24V. The output voltage value is exactly doubles the value of input voltage due to fact that the duty ratio value of the PWM signal is 50 percent of the time period applied to the MOSFET switch.

4.2. Boost Converter Closed-Loop Design

In this section, the input voltage controlled closed-loop design of the boost converter is explained. Since, the boost converter is used for the PV application and P&O MPPT and HPC algorithms are applied therefore the input side voltage needs to be controlled by the duty ratio values.

4.2.1. MATLAB Simulations

For the closed-loop design in MATLAB, the averaged state-space equations derived in (4.9)-(4.12) are need to be linearized by implementing the partial derivation of all the averaged variables. The resultant expressions are written in (4.15)-(4.18).

$$\frac{d\hat{\imath}_L}{dt} = \frac{1}{L}\hat{v}_C + \frac{r_C}{L}\hat{\imath}_{in} - \frac{(r_C + r_L + Dr_{sw} + D'r_d)}{L}\hat{\imath}_L - \frac{D'}{L}\hat{v}_o - \frac{((r_{sw} + r_d)I_{in} - V_o - V_d)}{L}\hat{d} \tag{4.15}$$

$$\frac{d\hat{v}_C}{dt} = \frac{1}{C}\hat{\imath}_{in} - \frac{1}{C}\hat{\imath}_L \tag{4.16}$$

$$\hat{v}_{in} = \hat{v}_C - r_C\hat{\imath}_{in} + r_C\hat{\imath}_L \tag{4.17}$$

$$\hat{\imath}_o = D'\hat{\imath}_L - I_L\hat{d} \tag{4.18}$$

$$R_1 = r_C + r_L + Dr_{sw} + D'r_d \tag{4.19}$$

The linearized state-space equations can be expressed in matrix form (4.20).

$$\frac{dx}{dt} = \boldsymbol{A}x + \boldsymbol{B}u\ ,\ y = \boldsymbol{C}x + \boldsymbol{D}u \tag{4.20}$$

Whereas;

$$x = \begin{bmatrix} \hat{\imath}_L \\ \hat{v}_C \end{bmatrix},\quad u = \begin{bmatrix} \hat{\imath}_{in} \\ \hat{v}_o \\ \hat{d} \end{bmatrix},\quad y = \begin{bmatrix} \hat{v}_{in} \\ \hat{\imath}_o \end{bmatrix}$$

The resultant coefficient matrices A, B, C and D are in (4.21) and (4.22).

$$\begin{bmatrix} \frac{d\hat{\imath}_L}{dt} \\ \frac{d\hat{v}_C}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_1}{L} & \frac{1}{L} \\ -\frac{1}{C} & 0 \end{bmatrix}\begin{bmatrix} \hat{\imath}_L \\ \hat{v}_C \end{bmatrix} + \begin{bmatrix} \frac{r_C}{L} & -\frac{D'}{L} & \frac{V_o + V_d}{L} \\ \frac{1}{C} & 0 & 0 \end{bmatrix}\begin{bmatrix} \hat{\imath}_{in} \\ \hat{v}_o \\ \hat{d} \end{bmatrix} \tag{4.21}$$

$$\begin{bmatrix}\hat{v}_{in}\\ \hat{\imath}_o\end{bmatrix}=\begin{bmatrix}-r_C & 1\\ D' & 0\end{bmatrix}\begin{bmatrix}\hat{\imath}_L\\ \hat{v}_C\end{bmatrix}+\begin{bmatrix}r_C & 0 & 0\\ 0 & 0 & -I_{in}\end{bmatrix}\begin{bmatrix}\hat{\imath}_{in}\\ \hat{v}_o\\ \hat{d}\end{bmatrix} \tag{4.22}$$

However, the equation (4.20) can also be written in Laplace transform as;

$$\begin{aligned} sX &= \boldsymbol{A}X + \boldsymbol{B}U \\ Y &= \boldsymbol{C}X + \boldsymbol{D}U \end{aligned} \tag{4.23}$$

By further solving, the resultant equation obtained between the outputs to the input variables is stated in (2.24)

$$\begin{aligned} X &= (s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}U \\ Y &= (\boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D})U = \boldsymbol{G}U \end{aligned} \tag{4.24}$$

Whereas;

$$\boldsymbol{G}=\begin{bmatrix}Z_{in-o} & T_{oi-o} & G_{ci-o}\\ G_{io-o} & -Y_{o-o} & G_{co-o}\end{bmatrix} \tag{4.25}$$

The matrix $\boldsymbol{G}$ represents all the possible control transfer functions between input to output variables. The elements of the matrix are represented according to their actual meaning, for instance, the element Z_{in-o} is named as input to control impedance, since it represents the input voltage to input current transfer functions. However, since in this thesis work the input voltage needs to be controlled by the duty ratio, therefore the element present in the first row third column in $\boldsymbol{G}$ matrix matches the requirements. i.e. $\boldsymbol{G}(1,3) = G_{ci-o}$.

The extended MATLAB code with the addition of linearized state-space matrices and the control to input transfer function G_{ci-o} is illustrated below;

```
%-----Linearized state-space model------%

Req = rC+rL+D*rsw+D1*rD;      % equivalent resistance from (4.19)

A = [-Req/L 1/L; -1/C 0];                % The co-officiant
B = [rC/L -D1/L (Uin+uD)/L; 1/C 0 0];    % matrices derived
C1 = [-rC 1; D1 0];                      % in equations
D2 = [rC 0 0; 0 0 -Iin];                 % (4.21) & (4.22)

sI = [s 0; 0 s];              % unity matrix
G = (C1*inv(sI-A)*B)+D2;      % the G matrix (4.24) & (4.25)
Gci = G(1,3);                 % control to input transfer function
```

In MATLAB simulations, PI controller G_C is used along with the G_{ci-o}. The PI controller is included with a DC-gain, two poles and two zeros and the transfer function is stated in symbolic form in (4.26)

$$G_c = \frac{K_{dc}\left(\frac{s}{\omega_{zero}}+1\right)\left(\frac{s}{\omega_{zero}}+1\right)}{s\left(\frac{s}{\omega_{pole}}+1\right)\left(\frac{s}{\omega_{pole}}+1\right)} \tag{4.26}$$

For the input voltage control mechanism, the cross-over frequency must be lower than the resonance frequency of the control system. Therefore, the G_c controller's DC-gain, poles and zeros values are set to get the required results. The numerical values of gain, poles and zeros is stated in (4.27).

- $K_{dc} = 10^{-\frac{40,5}{-20}} = 105.92$ (4.27)
- $\omega_{z1} = 2\pi * 900Hz = 5655\ rad/s$
- $\omega_{z2} = 5655\ rad/s$
- $\omega_{p1} = 2\pi * 10kHz = 62832\ rad/s$
- $\omega_{p2} = 62832\ rad/s$.

The Matlab code for the PI controller and the resultant bode plot of both the control transfer functions G_{ci-o} and G_c is stated below.

```
%-------Designing the PI Control function------%

Kdc = 10^(-40.5/20);     % The PI controller
wp1 = 2*pi*900;          % control parameters
wp2 = wp1;               % as defined in
wz1 = 2*pi*10000;        % (4.26) &
wz2 = wz1;               % (4.27)
Gc = Kdc*(s/wz1+1)*(s/wz2+1)*(1/(s*(s/wp1+1)*(s/wp1+1)));

P = bodeoptions;         % converting the bode plot
P.FreqUnits = 'Hz';      % unit parameters from 'rad' to 'Hz'
bode(Gci*Gc,P); grid;    % combine bode plot of the 'Gci-o' & 'Gc'
```

The resultant bode plot is shown in figure 4.6

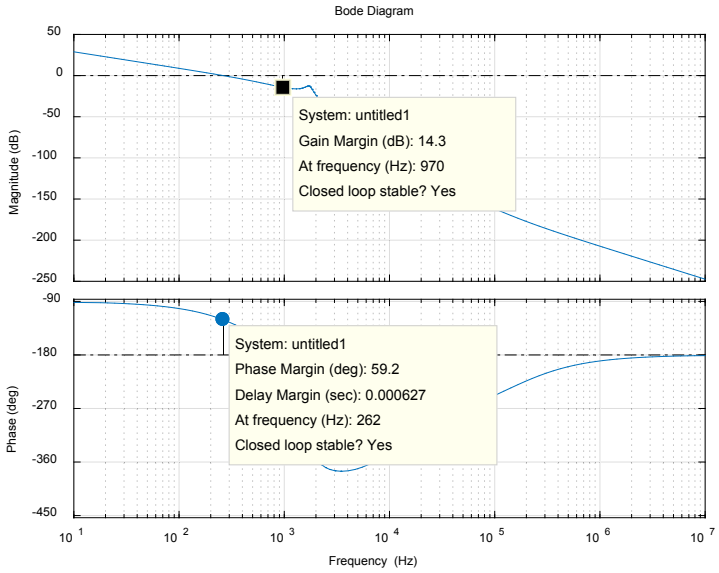
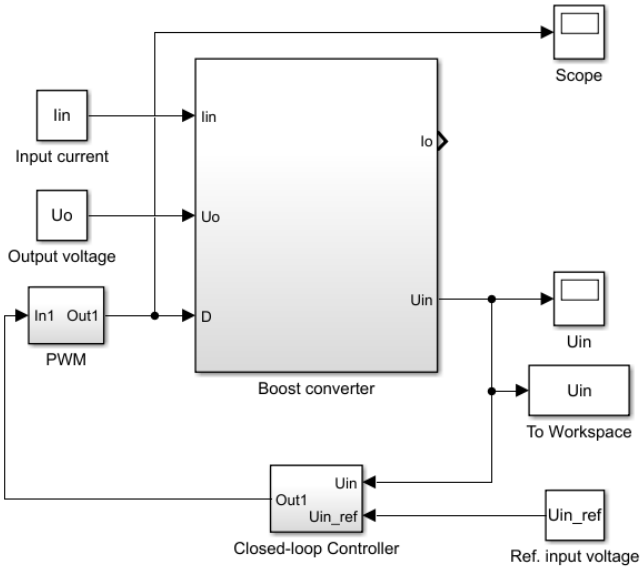
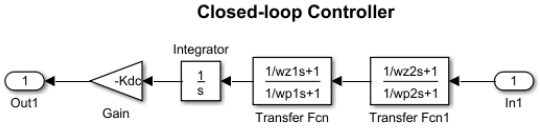


Figure 4.6. *The bode plot of the input voltage control transfer functions* $\boldsymbol{G_{ci-o}}$ & $\boldsymbol{G_c}$

Since the closed-loop input voltage control test is being performed, therefore, it needs to change the boost converter Simulink model shown in figure 4.4 (b) from open loop to closed loop settings with the addition of reference input voltage block. The generated input voltage is feed backed with the Reference input voltage to generate the error. Figure 4.7 depicts the Simulink model of the boost converter with closed-loop test.



(a)



(b)

Figure 4.7. *Simulink model of the boost converter with closed-loop control parameters*

Finally, the desired input voltage waveform obtained from Simulink model is stated in figure 4.8. There are a few spikes in the initial voltage values due to the transient time of the controlling mechanism. The complete closed loop MATLAB mfile code is written in Appendix A.

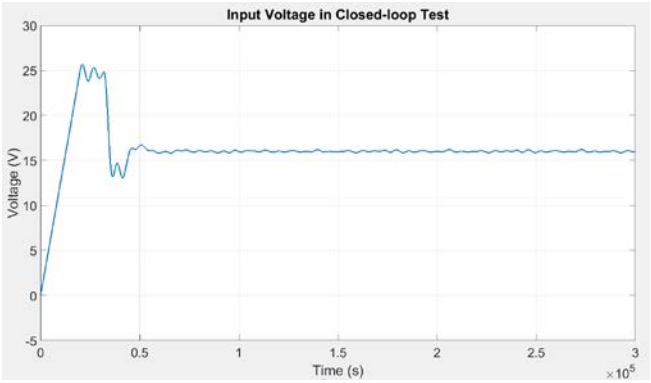


Figure 4.8. *The input voltage waveform with the closed-loop test*

4.2.2. CCS Closed-loop Test for MPPT

The closed-loop control of the boost converter for the MPPT algorithm needs to measure the input voltage and current values along with the continuous PWM generation from DSP. Therefore, some additional functions needs to include in the CCS code.

Header Files

```
#include "driverlib/adc.h"
```

The above header line of the code needs to request the access the ADC function from the `driverLib` library, since the ADC function is used for the measurement of input voltage and current.

External Functions Declaration

```
void ADC();
void MPPT();
void discrete();
```

These are the functions written outside the main function and compiler runs them only when they are called inside the main function. Above lines are for the declaration of the ADC, MPPT and the discretization functions, however, the actual body of these functions is written after the main function.

Defining Global Variables

```
uint32_t ADC_value[2];
float meas_voltage[2];
float meas_crnt;
```

```
float meas_pwr;
float count_adc = 0;
float count_mppt = 0;
```

Above lines are for the declaration of float type to the variables used in ADC processing. Unsigned 32-bit array `ADC_value[2]` is used to receive data from the GPIO pins configured for ADC. `meas_voltage[2]`, `meas_crnt` and `meas_pwr` are defined to store the calculated values of input voltage, current and power. However, `count_adc` and the `count_mppt` variables define the sampling time for the ADC process and the MPPT respectively.

```
float pwr[2];
float PV_pwr[2];
float e_0 = 4.2e-3;
float u_1 = -0.9958;
```

These variables are defined for the discretization of the low pass filter applied at the calculated input power. `PV_pwr[2]` array stores the filtered input power value. The error `e_0 = 0.0042` and the input `u_1 = -0.9958` values are calculated in MATLAB by applying the Backward Discretization method explained in [27].

Peripheral Enabling

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
```

In the above lines of code, ADC module 0 and the GPIO port E peripherals are enabled respectively from the `driverLib` directory.

ADC Configuration

```
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_3);
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_1);
```

The above two lines defines the GPIO pins PE3 for the input voltage measurement and the PE1 for the input current measurement respectively.

```
ADCSequenceConfigure(ADC0_BASE, 1, ADC_TRIGGER_PROCESSOR, 0);
```

This line configures the ADC mode 0 with the sample sequencer 1 and triggers the ADC processor with highest priority.

```
ADCSequenceStepConfigure(ADC0_BASE, 1, 0, ADC_CTL_CH0);
ADCSequenceStepConfigure(ADC0_BASE, 1, 1, ADC_CTL_CH2|ADC_CTL_IE|ADC_CTL_END);
```

These two lines defines the ADC sampling process. Channel0 `CH0` is dedicated with the PE3 pin and cannel2 `CH2` is for receiving data from PE1 as defined in table 2.2. However, `ADC_CTL_IE|ADC_CTL_END` instruction set is to end the ADC sampling process.

```
ADCIntClear(ADC0_BASE, 1);
ADCSequenceEnable(ADC0_BASE, 1);
```

The above line of these two code lines clear the ADC interrupt register before enabling the ADC peripheral in the `driverLib`. It is always a good practice to clear the ADC before enabling to avoid receiving any initial garbage values. However, the bottom line enables the ADC mode 0 sequencer 1 for the sampling.

ADC Function Code

```
void ADC(void)
{
      ADCProcessorTrigger(ADC0_BASE, 1);
      while(!ADCIntStatus(ADC0_BASE, 1, false))
      {}

      ADCSequenceDataGet(ADC0_BASE, 1, ADC_value);
      meas_voltage[0] = (3.3/4096)*10* ADC_value[0];
      meas_crnt = (((3.3/4096)*ADC_value[1]-2.500)*15.1515);
      meas_pwr[0] = meas_voltage[0]*meas_crnt;
}
```

The first line in the ADC function triggers the ADC0 sequencer 1 to start the process of receiving the data from PE3:CH0 and PE1:CH2. Whereas, the while loop is to let the compiler wait until the sampling process completed. Interrupt register triggering indicates the completion of the process.

The first line of the code segment written above defines the `ADC_value` variable that stores the ADC data received at CH0 and CH2. Since the `ADC_value` is an array of two variables `ADC_value[0]` and `ADC_value[1]`, therefore, system automatically allocates the first variable for the first channel `CH0` and second variable for the second channel `CH2`. However, rest of the three lines are for the calculation of the input voltage, current and the Power values from the ADC received data. The calculations for the voltage measurement are according to the method explained in detail in the section 2.6.1 above. Whereas, since in this thesis work, ACS712-30A is used for current sensing and according to the data sheet, its sensing resolution is 66mV against 1A current. Moreover, the varying output voltage range is from 2.5V to 5V. Therefore, the expression can be derived for the current measurement is stated in (4.28).

$$Meas.\ Current = \frac{3.3}{4096} \times (sensed\ voltage - 2.5) \times \frac{1}{0.066} \tag{4.28}$$

MPPT Function Code

```
void MPPT(void)

{
     if((PV_pwr[0] - PV_pwr[1]) == 0)
     {}
     else
     {
          if((PV_pwr[0] - PV_pwr[1]) > 0)
           {   if((meas_voltage[0] - meas_voltage[1]) > 0)
                  { duty = duty - 0.3; }
               else
                  { duty = duty + 0.3; }
           }
          else
           {   if((meas_voltage[0] - meas_voltage[1]) > 0)
                   { duty = duty + 0.3; }
               else
                   { duty = duty - 0.3; }
           }
     }
```

The above segment of the MPPT function follows the P&O MPPT algorithm illustrated in flowchart of figure 2.4. Where `PV_pwr[0]` and `meas_voltage[0]` represents the present values of the measured power and voltage. However, `PV_pwr[1]` and `meas_voltage[1]` represents the measured power and voltage values used in previous MPPT sampling cycle.

```
     if(duty < 30)
            {
               duty = 30;
            }
     if(duty > 90)
            {
               duty = 90;
            }
}
```

These remaining lines of the MPPT function limit the range of the operating point on the P-V characteristic curve to ensure the safety and stability of the PV system.

Discretization Function Code

```
void discrete(void)
{
     pwr[0] = e_0*meas_pwr[0] - u_1*pwr[1];
```

```
        pwr[1] = pwr[0];
        PV_pwr[0] = pwr[1];
}
```

This is an external function acts as a low pass filter to stabilize the measured power value. The `PV_pwr[0]` variable stores the filtered power and then used for MPPT algorithm.

Infinite While() Loop

```
while (1)
      {
            if(count_adc == 1000)
            {
                  ADC();
                  discrete();
                  count_adc = 0;
            }
```

The above segment of the while loop comprised of calling the `ADC()` and `discrete()` function. However, the `if` statement defines the sampling frequency for the ADC and consequently the discretization processes. Since the system clock runs at 40MHz and `count_adc` counts to 1000 times then allows the compiler to call the ADC function, therefore the ADC frequency is;

$$ADC\ Sample\ Frequency = \frac{1}{40MHz} \times 1000 = 250mSec = 40kHz$$

```
            if(count_mppt == 1000000)
            {
                  MPPT();
                  PV_pwr[1] = PV_pwr[0];
                  meas_voltage[1] = meas_voltage[0];
                  count_mppt = 0;
            }
```

In these code lines, the `MPPT()` function is called along with the transferring of the present power and voltage values to the variables having the previous values. The `if` condition describes the MPPT sampling frequency. As depicted, the `count_mppt` counts to 1000K times the system clock therefore;

$$MPPT\ Sample\ Frequency = \frac{1}{40MHz} \times 10^6 = 0.5Sec = 2Hz$$

```
            count_adc++;
            count_mppt++;
      }
```

Since the compiler is running an infinite loop therefore, these variables `count_adc` and `count_mppt` are incremented by one on each system clock cycle. The MPPT experimental results are presented in detail in the chapter 5 of the thesis.

4.2.3. Implementation of Limit Power Algorithm

Since the closed-loop ADC function and the MPPT algorithum are successfully implemented, the next implementation technique of the thesis is Limit-Power or HPC algorithm. Therefore, the CCS code needs to be developed accordingly to implement the HPC on DSP board. Following are few additional changes in the CCS code that needs to be expalined in detail.

External Functions Declaration

```
void lim_p();
```

For the HPC algorithm, an additional external function is declared along with the ADC,MPPT and the discrete functions.

HPC Function Code

```
void lim_p (void)

{
     if((PV_pwr[0] - PV_pwr[1]) == 0)
     {}
     else
     {
          if((PV_pwr[0] - PV_pwr[1]) > 0)
           {   if((meas_voltage[0] - meas_voltage[1]) > 0)
                  { duty = duty + 0.3; }
               else
                  { duty = duty - 0.3; }
           }
          else
           {   if((meas_voltage[0] - meas_voltage[1]) > 0)
                   { duty = duty - 0.3; }
               else
                   { duty = duty + 0.3; }
           }
     }
     if(duty < 30)
            {
               duty = 30;
            }
     if(duty > 90)
```

```
            {
                duty = 90;
            }
}
```

The programming logic behind the HPC is the same as in the MPPT function. However, the HPC works exactly in the opposite way to detrack the operating point away from the MPP point on P-V curve. The above code of the `lim_p()` explains the idea of HPC.

Infinite While() Loop

```
if(count_mppt == 1000000)
          {
               if (PV_pwr[0] >= 12)
               { lim_p(); }
               else
               { MPPT(); }
               PV_pwr[1] = PV_pwr[0];
               meas_voltage[1] = meas_voltage[0];
               count_mppt = 0;
          }
```

The `if else` condition shown in the above instruction set explains that in one sampling cycle, either MPPT function or HPC function runs. However, In the while loop, the `lim_p()` function is within double cascaded `if` conditions. The first condition defines the sampling frequency, whereas, the inner condition states the value of the P_{limit} for the system. When the measured power `PV_pwr[0]` value is less than the P_{limit} the MPPT function is called, however, when the `PV_pwr[0]` reaches the limit or become greater than the limit power, compiler calls the limit power function. In this case, the P_{limit} value in 12W. The complete CCS code is written in the Appendix B

5. SIMULATIONS AND RESULTS

In chapter 4 of this thesis, the working principles of boost converter with MPPT and HPC algorithms, MATLAB simulations and detailed description of the CCS code is presented. However, in this chapter the experimental results of implementing the MPPT algorithm with different irradiance conditions and the HPC algorithm are described.

5.1. Experimental Setup

The prototype circuit used for experimental results in shown in figure 3.6, whereas, the operating parameters of the boost converter and the PV module are defined in the tables 3.1 and 3.2 respectively. In addition, 24V battery load is connected to the load side of the boost converter. Figure 5.1 depicts the overall experimental setup in PE laboratory including the PV module, the boost converter circuitry, the Tiva Launchpad uC and the battery bank. The lab's temperature conditions are normal, i.e. room temperature 25°C, for all the experimental results presented below.

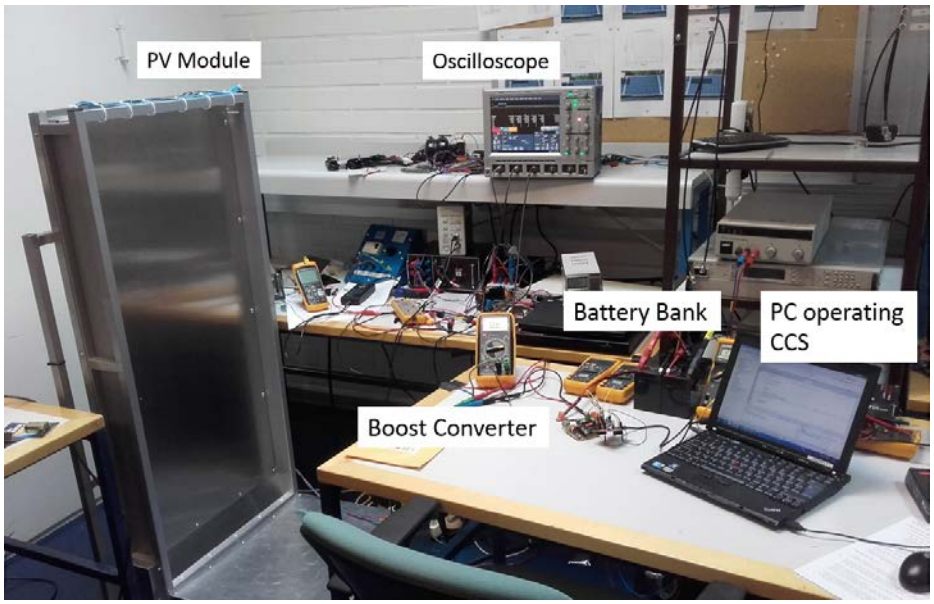
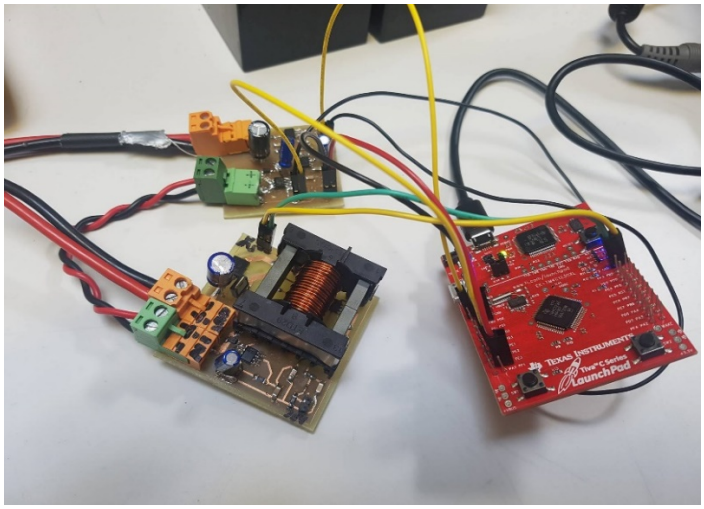


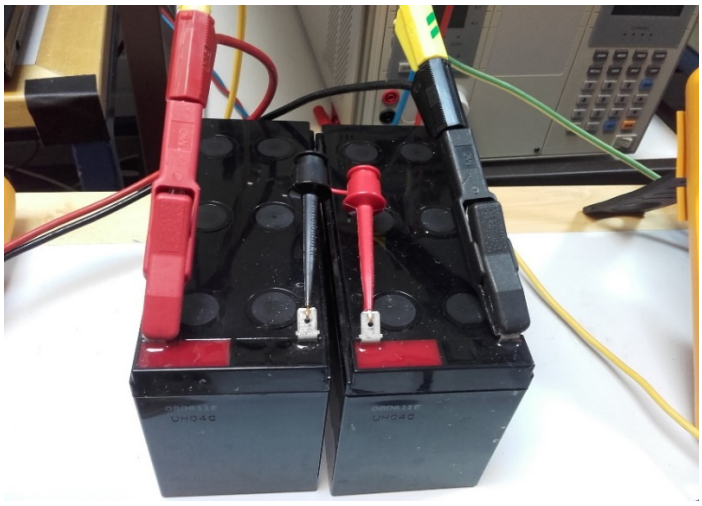
Figure 5.1. *Experimental setup in the PE laboratory*

However, the oscilloscope used to get the resultant waveforms is *LeCroy WaveRunner 104MXi* 1GHz Oscilloscope as shown in picture 5.2 (a). Figure 5.2 (b) presents the boost converter and the measurement circuitry connected with Tiva Launchpad through jumper wires. The battery bank includes two sets of 12V lead-acid batteries connected in series as illustrated in figure 5.2 (c).

(a)



(b)



(c)

Figure 5.2. *The various project modules take part in the lab experiments, (a) the oscilloscope presenting the results, (b) the experimental setup of the Launchpad with the boost converter and the measuring circuitry, (c) the battery bank used as a load.*

5.2. Maximum Power Point Tracking

Results with 800-900W/m^2

The PV module used in this thesis work is 20W rated power and according to [14], [15] the expected MPP value of the module must be 76 percent of the rated power. In the present case, the MPP is around 16.5W with 800-900W/m^2 light intensity. Figures 5.1 represent the PV power generated with the constant light intensity of around 900 W/m^2.

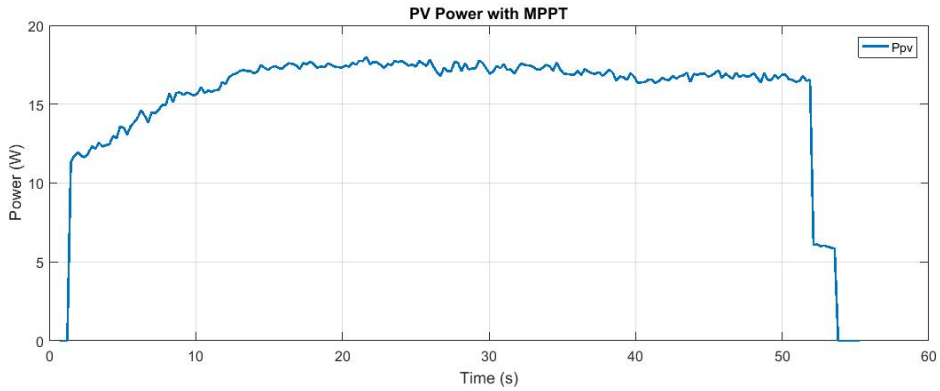


Figure 5.3. *The PV power generation waveform at 800-900 W/m^2 light intensity*

As depicted from the figure 5.3, the maximum power is around 17W which is quite close to the expected maximum power at 900 W/m^2 for the PV module. However, the initial P_{pv} values are small due to fact that the system requires some time to track the MPP.

Figure 5.4 presents the input voltage V_{pv} current I_{pv} and the Power P_{pv} waveforms in a single figure to get the better understanding of the MPPT algorithm. As illustrated in the figure 5.4, the voltage V_{pv} values are high and the power P_{pv} value is small that means that initially operating point is at the right side of the P-V curve and then tracked to the MPP. Figure 5.5, proves this statement of MPP tracking, since the waveform is plotted with the voltage on the x-axis and the power on the y-axis. As shown, the operating point initially remains on the right side and then tracked to MPP.

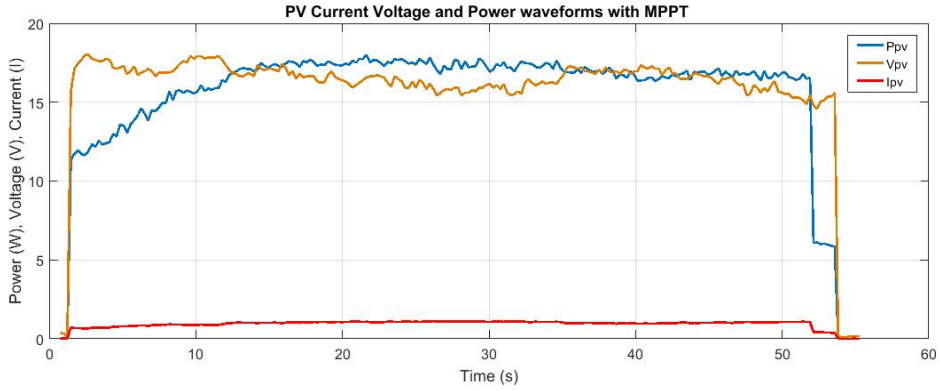


Figure 5.4. *PV current, voltage and power waveforms at light intensity of 800-900 W/m^2*

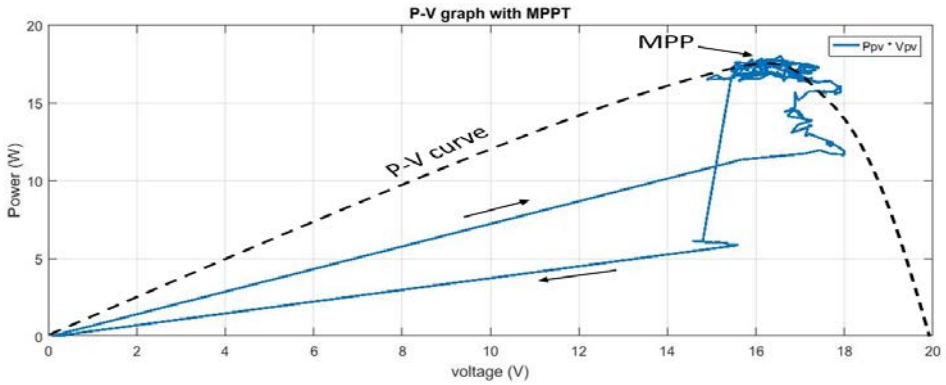


Figure 5.5. *The waveform of the operating point with MPPT*

Results with 500-600W/m^2

In this section, the experiment is conducted at low irradiance value of around 500-600W/m^2 to illustrate the case of insufficient light that might be due to cloudy weather or winter times e.g. Figure 5.6 states the experimental waveform of PV power generated under a constant light intensity of 500-600W/m^2.

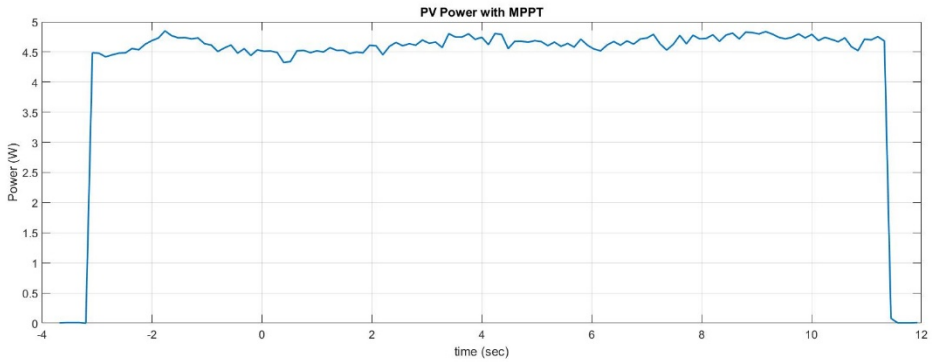


Figure 5.6. *PV power generation waveform at 500-600 W/m^2 light intensity*

As shown in figure 5.6, the power value is around 4.7 W due to low irradiance value of 500-600W/m^2. Initially tracker takes two to three seconds to track the MPPT because the power value is rising. However, due to quite low irradiance value the available power is not more than almost 5 Watts and that is why tracker reaches MPP in a short time.

Figure 5.7 presents the PV voltage, current and power waveforms on a single plot with the constant light intensity of 500-600W/m^2. By comparing figure 5.7 with figure 5.4 above, it is quite clear that the PV current value is very small and consequently the PV Power as well compared with that of figure 5.4. However, the voltage values in both the waveforms remains more or less the same. This proves that the by changing the irradiance values, the PV current effects more rather than PV voltage.

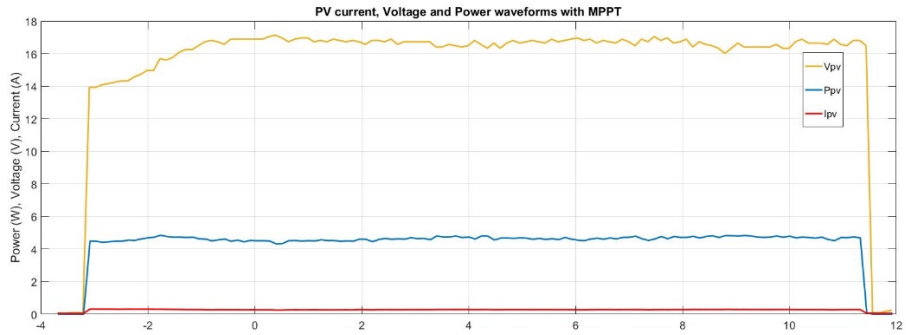


Figure 5.7. *PV current, voltage and power waveforms at light intensity of 500-600 W/m²*

Further, to ensure that the PV power obtained at low intensity value is the MPP power, the PV power and voltage is plotted in figure 5.8 as shown. It is quite clear from the figure that the operating point remains almost all the time at MPP which ensures that the PV generated power plotted in figure 5.6 is maximum power at available irradiance value.

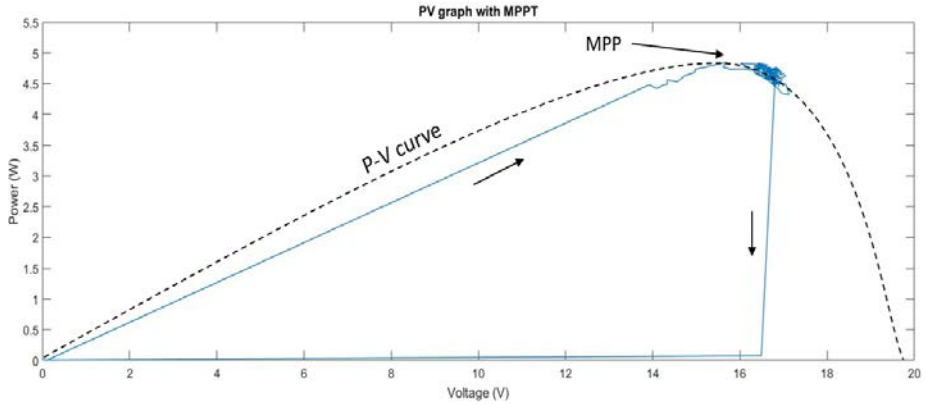


Figure 5.8. *The waveform of the operating point with MPPT*

The experimental results shown above proves that the boost converter prototype, designed in this thesis, is capable of tracking MPP at each and every irradiance values.

5.3. Hybrid Power Control

The HPC algorithm is tested in this thesis with the continuously varying light intensity conditions from 500 W/m^2 to 800 W/m^2 irradiance values. Figure 5.9 states two different waveforms plotted on the equally varying irradiance conditions within equal interval of time. However, the waveform labeled with "MPPT" is plotted when the PV system is operating with MPPT algorithmic mode and the one labeled with "HPC" is plotted when the PV system is operating with power limiting algorithm. The power limit P_{limit} is set to 9 Watts. As seen from the figure, initially when the PV generated power P_{pv} value is less than the P_{limit} value, both the ware forms appears to be almost identical. However, when the irradiance value increases to the point where P_{pv} becomes equal to the P_{limit} then the Limit-power curve started moving horizontally, i.e. the PV system tries to limit the P_{pv} to the same value, however, the MPPT curve kept on moving upwards in response

to the increasing irradiance value and ultimately reaches the maximum power value. The waveforms are plotted during the time period of 120 seconds.

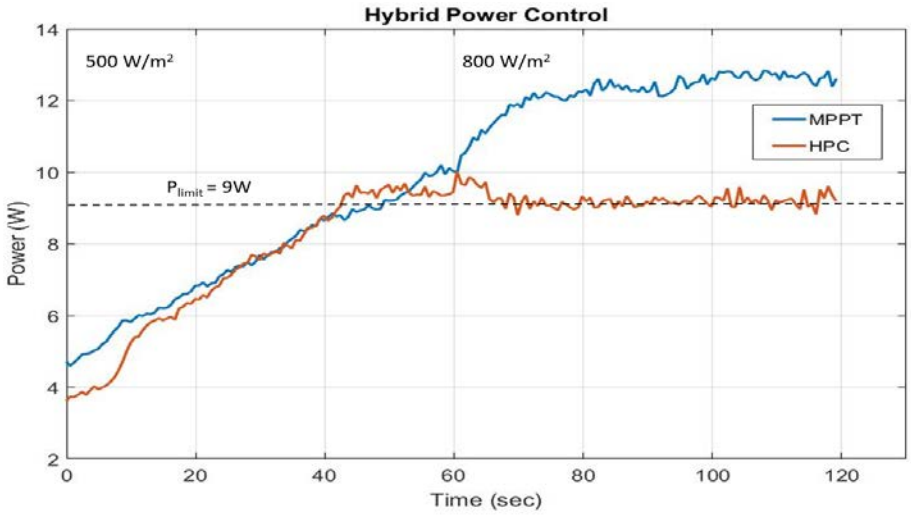


Figure 5.9. *Comparison of the hybrid power controlled waveform and the maximum power point tracked waveform plotted at identical climatic conditions*

6. CONCLUSION AND FUTURE WORK

6.1. Final Conclusion

The advancement of the grid-connected PV systems on the commercial level in last few decades has increased the importance of the issues related to the PV systems. Two of the vital issues are discussed in this thesis work. It is examined that the solar energy to electricity conversion efficiency can be increased significantly by implementing the Maximum Power Point Tracking Algorithm MPPT. The implementation of MPPT is also explained in detail in this thesis work. Moreover, another major issue is the large and rapid PV generated power fluctuations due to varying climate conditions even on a normal sunny day which directly effects the grid voltage and frequency. As a consequence the safety and stability of the utility grid starts damaging due to increasing amount of the grid connected renewable systems. Therefore, it is observed that by limiting the PV generated power to a certain level can improve the overall stability of the PV power and the magnitude of the peak to peak PV power fluctuations become smaller. The implementation of Limit-power algorithm is discussed in detail in the thesis and the results are examined. However, since the thesis relates to the practical work of designing of the boost converter as well, therefore the detailed explanation of the software and hardware designing procedures is also included in the thesis. Moreover, the TM4C123G Tiva Launchpad DSP has played a vital role during the whole thesis work starting from the testing of the boost converter open-loop and closed-loop tests, then the implementation of the MPPT and Limit-power algorithms. Therefore, the brief demonstration of the DSP architecture and some of its useful functions, e.g. ADC, PWM…, utilized in the thesis work are explained

6.2. Basic Learning and Issues

The author has learnt many of the quite useful skills during the thesis work, most importantly, the learning of the new coding language for the DSP utilization. CCS software uses the C programming language for the code compilation and to program the DSP. However, the PCB designing experience in terms of software as well as hardware was quite beneficial. PADS Logic and Layout interface is really helpful for designing PCB. In addition, the inductor designing and the components soldering processes was quite challenging but valuable for the author.

Since the implementation and working of hardware is always a challenging task compared with that of the software. Therefore, it takes the author several months to make the hardware work properly, although the software and coding procedures were completed months before the successful hardware implementation. The most annoying challenge author has faced, was related to the PCB hardware designing. The issue was with the

improper common grounding of the PCB which affected the results of various components attached especially the MOSFET driver, and it took several weeks to configure the fault. Moreover, while writing the code for the ADC function, the issue was the incorrect configuration of channels defined for receiving the data with the DSP Port pins. The issue was due to incomplete or careless reading of the DSP datasheet because the solution was written there. Anyhow, it took couple of weeks to get rid of the issue. However, there are certain small issues related to the MPPT and Limit-power implementation, from which author had learnt a lot.

6.3. Expected Future Research

Solar technology, especially the grid connected PV systems are presently the main focus area of research. With the increasing integration of the PV to the grid, the grid stability challenges become more vital. Limit-power is one of the best solutions for the improved grid stability, therefore, it needs to be further develop and improve. The publication referenced at [28] presents quite similar concept related to the limit-power algorithm, i.e. the Ramp-Power Control RPC. The RPC basically gradually increasing the PV power with a constant ramp rate depending upon the grid requirement instead of limiting. This concept of ramp power is quite useful as well for the safety of the grid. Therefore, the limit-power concept might adopt quite different procedures and ways in the future to improve the power quality of the PV systems.

7. REFERENCES

[1] A. Demirbas, "Electricity via Unconventional Methods," *Energy Exploation & Exploitation,* (2006), 24(1-2): 131-138.

[2] Tuomas Messo, "Factors Affecting Stable Operation of Grid-Connected Three-Phase Photovoltaic Inverter," *PhD Thesis Report at Tampere University of Technology,* 2014.

[3] Observ'ER et al., "Worldwide Electricity Production from Renewable Energy Sources," *Stats and Figure series 2013 edition,* 2013.

[4] U.S. department of Energy. The History of Solar, *Energy Efficiency and Renewable Energy*

[5] Jessika Toothman & Scott Aldous. (2000). *How Solar Cells Work,* [Online] Available: http://science.howstuffworks.com/environmental/energy/solar-cell.htm

[6] A.Attou et al., "Photovoltaic Power Control Using MPPT and Boost Converter," *Balkan Journal of Electrical & Computer Engineering* 2(1): 23-27, 2014.

[7] R. F. Coelho & D. C. Martins, "An Optimized Maximum Power Point Tracking Method Based on PV Surface Temperature Measurement." *Chap: 4, published by INTECH,* 2012.

[8] M. A. Green, et al., "Solar cell efficiency tables (version 35)", *Progress in Photovoltaics: Research and Applications*, (18)144–150, 2010.

[9] A. Gaur and G. N. Tiwari, "Performance of Photovoltaic Modules of Different Solar Cells." *Research Article from the Journal of Solar Energy,* 2013.

[10] Choi, U. M., et al., "Power electronics for renewable energy systems: Wind turbine and photovoltaic systems." *Renewable Energy Research and Applications (ICRERA), Int. IEEE Conf.* 2012

[11] Dr. George S. Cheng. "Designing a Scalable On/Off-Grid Solar Power System." *North America Clean Energy,* vol. 2, 2016.

[12] Liz Osborn. *"Sunniest Places and Countries in the World."* [Online]. Available: http://www.currentresults.com/Weather-Extremes/sunniest-places-countries-world.php (12 May, 2017)

[13] Green, Martin A., et al. "Solar cell efficiency tables (Version 45)." *Progress in photovoltaics: research and applications* (23.1) 1-9, 2015.

[14] D. Sera, T. et al., "Improved MPPT Algorithms for Rapidly Changing Environmental Conditions," *Power Electronics and Motion Control Conf., EPE-PEMC 12th Int. IEEE* 1614-1619, 2006.

[15] Hohm D.P., Ropp ME. "Comparative Study of Maximum Power Point Tracking Algorithms Using an Experimental, Programmable, Maximum Power Point Tracking Test Bed." *PV specialist conf.* 28(15-22), 1699-1702*,* 2000.

[16] K. Nanshikar, A. Desai. "Simulation of P & O Algorithm using Boost Converter." *Nitte Conf. Advances in Electrical Engineering NCAEE-2016* vol. 4, pp 130-135, 2016.

[17] Y. Yang, et al., "A Hybrid Power Control Concept for PV Inverters with Reduced Thermal Loading." *IEEE Transactions on Power Electronics,* 29(12), 6271-6275, 2014.

[18] D. Rosenwirth & K. Strubbe, "*Integrating variable renewables as Germany epands its grid,*" [Online] Available: http://www.renewableenergyworld.com/, 2013. (14 May, 2017)

[19] H. Gaztanaga, et al., "Enhanced experimental PV plant grid-integration with a MW Lithium-Ion energy storage system," *in Proc. of ECCE*, 1324-1329, 15-19, 2013.

[20] Tiva™ TM4C123GH6PM Microcontroller Data Sheet. (2014) [Online] Available: http://www.ti.com/lit/ds- /symlink/tm4c123gh6pm.pdf. (15 May, 2017)

[21] Scott Bland. (2015) "*TM4C123G Launchpad Workshop Series 5 of 15: ADC12.*" [Online] Available: https://training.ti.com/tm4c123g-launchpad-workshop-series-5-15-adc12 (15 May, 2017)

[22] Scott Bland. (2015) *"TM4C123G Launchpad Workshop Series 15 of 15: PWM."*[Online] Available: https://training.ti.com/tm4c123g-launchpad-workshop-series-15-15-pwm (16 May, 2017)

[23] Wikipedia, The free encyclopedia. (May, 2017) *"Code Composer Studio"* [Online] Available: https://en.wikipedia.org/wiki/Code_Composer_Studio (17 May, 2017)

[24] Aapo Aapro. (2016). *"DEE-34037 Design Course in Power Electronics."* [Online], Available: http://webhotel2.tut.fi/units/set/opetus/kurssit/Materaalisivut/DEE_34037/Lecture%203%20-%20Schematics%20and%20Inductor%20Design.pdf (17 May, 2017)

[25] Seppo Valkealahti. (2017). *"DEE-53117 Solar Power Systems."* [Online], Available: http://webhotel2.tut.fi/units/set/opetus/kurssit/Materiaalisivut/DEE-53117/2%20Solar%20radiation.pdf. (19 May, 2017)

[26] Coral Saab. (2013). *"It's getting hot in here"* [Online], Available: https://csiro-solarthermal.wordpress.com/author/carolsaab80/ *Solar Technology blog, National Solar Energy Centre (NSEC)*. (22 May, 2017)

[27] Aapo Aapro. (2016). "DEE-34037 Design Course in Power Electronics." [Online] Available: http://webhotel2.tut.fi/units/set/opetus/kurssit/Materiaalisivut/DEE_34037/Lecture%204%20-%20Controller%20discretization%20and%20DSP%20implementation.pdf. (22 May, 2017)

[28] Y. Yang, et al., "A Cost-Effective Power Ramp-Rate Control Strategy for Single-Phase Two-Stage Grid-Connected Photovoltaic Systems." *IEEE Transactions on Power Electronics,* 29(12), 6271-6275, 2015.

[29] eurostat Statistics Explained, (March 2017), *"Energy from renewable sources",* [Online], Available: http://ec.europa.eu/eurostat/statistics-explained/index.php/Energy_from_renewable_sources#Installed_capacity_for_renewable_electricity_generation (08 June, 2017)

8. APPENDIX A: MATLAB CODE

```
%Open-loop control of boost converter
%PV panel spec: 20W, Voc = 21.5V, Isc = 1.25A
clear all
clc

%------Converter Design Parameters------%

fs = 150e3;      % MOSFET Switching frequency
Ts = 1/fs;       % Time period for one switching cycle
Uin = 16;        % Input voltage
Uo = 24;         % Output voltage
Iin = 1.25;      % Input current
L = 84.19e-6;    % Inductance value calculated in section 3.1.1
rL = 100e-3;     % inductor resistance given in the data sheet
rC = 10e-3;      % capacitor resistance given in the data sheet
rD = 10e-3;      % forward biased schottky diode resistance
uD = 0.5;        % schottky diode forward voltage drop
rsw = 70e-3;     % MOSFET drain-to-source conduction resistance
C = 100e-6;      % Capacitance value
s = tf('s');     % defining the laplace transfer function 's'

%-------Steady-state Duty Ratio-------%

D1 = (Uin - (rL + rsw)*Iin)/...    % duty ratio expression derived from
     (Uo + uD -(rsw -rD)*Iin);     % equation (4.14)
D =  1-D1;

%-----Linearized state-space model------%

Req = rC+rL+D*rsw+D1*rD;      % equivalent resistance from (4.19)

A = [-Req/L 1/L; -1/C 0];               % The co-officiant
B = [rC/L -D1/L (Uin+uD)/L; 1/C 0 0];   % matrices derived
C1 = [-rC 1; D1 0];                     % in equations
D2 = [rC 0 0; 0 0 -Iin];                % (4.21) & (4.22)

sI = [s 0; 0 s];              % unity matrix
G = (C1*inv(sI-A)*B)+D2;      % the G matrix (4.24) & (4.25)
Gci = G(1,3);                 % control to input transfer function

%-------Designing the PI Control function------%

Kdc = 10^(-40.5/-20);    % The PI controller
wp1 = 2*pi*900;          % control parameters
wp2 = wp1;               % as defined in
wz1 = 2*pi*10000;        % (4.26) &
wz2 = wz1;               % (4.27)
Gc = -Kdc*(s/wz1+1)*(s/wz2+1)*(1/(s*(s/wp1+1)*(s/wp1+1)));

P = bodeoptions;         % converting the bode plot
P.FreqUnits = 'Hz';      % unit parameters from 'rad' to 'Hz'
bode(Gci*Gc,P); grid;    % combine bode plot of the 'Gci-o' & 'Gc'
```

9. APPENDIX B: CCS CODE FOR DSP

```
// Maximum and Limit-Power Tracking Algorithms in PV Applications
// Supervised by:  Tuomas Messo
// Designed by:  Muhammad Ammar Ahmed

#include <stdint.h>
#include <stdbool.h>
#include "driverlib/gpio.h"
#include "driverlib/pin_map.h"
#include "driverlib/pwm.h"
#include "driverlib/adc.h"
#include "utils/uartstdio.c"


//--------------external functions declaration------------------//

void ADC();      // function for voltage ADC conversion
void MPPT();     // function for MPPT using perturb and observe
void lim_p();    // function for Limit-power algorithm
void discrete();        // Discretization function

//-----------------------global Variables-----------------------//

uint32_t ADC_value[2]; // 32-bit variables for ADC input
float meas_voltage[2];                // 32-bit variable to store measured voltage value from ADC
float meas_crnt;                      // 32-bit variable to store measured current   value from ADC
float meas_pwr[1];             // For measuring PV power
float pwr[2];
float PV_pwr[2];

float period = 133;            // 6.65usec --> (40Mhz / 2 pwm_divider / 133)
float duty = 67;        // 3.48usec with D=0.5

float count_adc = 0;    // For ADC sampling frequency
float count_mppt = 0;   // For MPPT sampling frequency

float e_0 = 0.0042;     // error value for discretization
float u_1 = -0.9958;    // input value for discretization


//====================== Main function ============================//
int main(void)
{
               //initial parameters to start the MPPT or lim-power operation
        pwr[1] = 15;
    duty = 67;

                                      //Set system crystal to 40MHz clock cycle frequency
SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_OSC_MAIN|SYSCTL_XTAL_16MHZ);

                                      //Configure PWM Clock divide system clock by 2
SysCtlPWMClockSet(SYSCTL_PWMDIV_2);    // PWM clock = 20MHz --> (system clock = 40MHz / 2)

                                          // Enable the peripherals used in the code.
SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);    // peripheral for the ADC function

SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM1);    // peripheral for the PWM function

SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);   // peripheral for the general purpose port-F pins dedicated for the PWM

SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);   // peripheral for the general purpose port-E pins dedicated for the ADC


//-----------------------PWM configurations-------------------//
               // Configure PF2 Pin as PWM mode 1
```

```
GPIOPinConfigure(GPIO_PF2_M1PWM6);
GPIOPinTypePWM(GPIO_PORTF_BASE, GPIO_PIN_2);

                // Configure PWM1 with PWM-generator 3
PWMGenConfigure(PWM1_BASE,PWM_GEN_3,PWM_GEN_MODE_DOWN|PWM_GEN_MODE_NO_SYNC);

                // Set the Period for 1 PWM cycle
PWMGenPeriodSet(PWM1_BASE, PWM_GEN_3, period);

                // Set PWM duty
PWMPulseWidthSet(PWM1_BASE, PWM_OUT_6,duty);

                // Enable the PWM mode 1 and generator 2
PWMGenEnable(PWM1_BASE, PWM_GEN_3);

                // Turn on the PWM Output pin
PWMOutputState(PWM1_BASE, PWM_OUT_6_BIT, true);


 //---------------------ADC Configurations------------------//

                // set PE2 pin for input to the voltage measurement
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_3);
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_1);

                // configured ADC mode 0 and sample sequencer 1 and trigger   ADC processor with highest
                priority
ADCSequenceConfigure(ADC0_BASE, 1, ADC_TRIGGER_PROCESSOR, 0);

                // configuration of sequencer 1 to collect data from ADC defined PE2 pin
ADCSequenceStepConfigure(ADC0_BASE, 1, 0, ADC_CTL_CH0);
ADCSequenceStepConfigure(ADC0_BASE,1,1,ADC_CTL_CH2|ADC_CTL_IE|ADC_CTL_END);

                // clear ADC interrupt status flag before ADC process
ADCIntClear(ADC0_BASE, 1);

                // enable ADC0 sequencer 1
ADCSequenceEnable(ADC0_BASE, 1);


//-------------------Infinite While Loop---------------//
while (1)
 {
PWMPulseWidthSet(PWM1_BASE, PWM_OUT_6, duty);    // PWM execution command

                if(count_adc == 1000)             // condition for sampling the ADC = 40kHz
                {
                  ADC();                          // function call for the ADC process
                  discrete();                     // function call for the discretization of meas. power

                  count_adc = 0;
                }

                if(count_mppt == 1000000)         // condition for sampling the MPPT or Lim-P = 2Hz
                {
                        if (PV_pwr[0] >= 12)      // if-else condition for the P-limit set
                        { lim_p(); }              // function call for the Lim-Power process

                        else
                        { MPPT(); }               // function call for the MPPT process

                        PV_pwr[1] = PV_pwr[0];
                        meas_voltage[1] = meas_voltage[0];

                        count_mppt = 0;
                }

                count_adc++;
                count_mppt++;
        }

}
//==================== main function ends ============================//
```

```
//--------------------External Functions--------------------------------//

void ADC(void)
{
  ADCProcessorTrigger(ADC0_BASE, 1);           // ADC process trigger to start conversion process

  while(!ADCIntStatus(ADC0_BASE, 1, false))    // Wait for conversion to be completed.
  {}

               //data received from pin PE2 discussed above is then stored in "ADC_value" named array.
               //The depth of sequencer determines the size of array.
               //Since we are using sequencer 3 therefore the array size is 1.
  ADCSequenceDataGet(ADC0_BASE, 1, ADC_value);


  meas_voltage[0]=(3.3/4096)*10* ADC_value[0];         // The output of ADC is scaled to meas. volts

  meas_crnt=(((3.3/4096)*ADC_value[1]-2.500)*15.1515); // The output of ADC is scaled to meas. current

  meas_pwr[0] = meas_voltage[0]*meas_crnt;             // meas. Power from meas. Current and voltage
}


void MPPT(void)
{
                                                 // implemented according to fig. 2.7
        if((PV_pwr[0] - PV_pwr[1]) == 0)
        {}
        else
        {
                if((PV_pwr[0] - PV_pwr[1]) > 0)
                 {   if((meas_voltage[0] - meas_voltage[1]) > 0)
                        { duty = duty - 0.3; }
                     else
                        { duty = duty + 0.3; }
                 }
                else
                 {   if((meas_voltage[0] - meas_voltage[1]) > 0)
                               { duty = duty + 0.3; }
                     else
                               { duty = duty - 0.3; }
                 }
        }

        if(duty < 30)
                  {
                        duty = 30;
                  }
        if(duty > 90)
                  {
                        duty = 90;
                  }

}


void lim_p(void)
{
                                                 // implemented according to fig. 2.7
        if((PV_pwr[0] - PV_pwr[1]) == 0)
        {}
        else
        {
                if((PV_pwr[0] - PV_pwr[1]) > 0)
                 {   if((meas_voltage[0] - meas_voltage[1]) > 0)
                        { duty = duty + 0.3; }
                     else
                        { duty = duty - 0.3; }
                 }
                else
                 {   if((meas_voltage[0] - meas_voltage[1]) > 0)
```

```
                        { duty = duty - 0.3; }
                    else
                        { duty = duty + 0.3; }
                }
        }

        if(duty < 30)
                {
                    duty = 30;
                }
        if(duty > 90)
                {
                    duty = 90;
                }

}


void discrete(void)
{

        pwr[0] = e_0*meas_pwr[0] - u_1*pwr[1];         // discretization equation from [27]

        pwr[1] = pwr[0];
        PV_pwr[0] = pwr[1];

}
```